

**SEGMENTAL DISCRIMINATIVE ANALYSIS FOR AMERICAN SIGN
LANGUAGE RECOGNITION AND VERIFICATION**

A Thesis
Presented to
The Academic Faculty

by

Pei Yin

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing, College of Computing

Georgia Institute of Technology
May 2010

SEGMENTAL DISCRIMINATIVE ANALYSIS FOR AMERICAN SIGN LANGUAGE RECOGNITION AND VERIFICATION

Approved by:

Thad Starner, Advisor
School of Interactive Computing, College
of Computing
Georgia Institute of Technology

Irfan Essa, Co-Advisor
School of Interactive Computing, College
of Computing
Georgia Institute of Technology

James M. Rehg, Co-Advisor
School of Interactive Computing, College
of Computing
Georgia Institute of Technology

Harley Hamilton
School of Interactive Computing, College
of Computing
Georgia Institute of Technology

Stan Sclaroff
Department of Computer Science
Boston University

Date Approved: April 2010

*To my family, especially my parents,
Weijie Yin and Yun Ge,
who made all of this possible,
for their endless encouragement and patience.*

ACKNOWLEDGEMENTS

I owe special thanks to many people, whose support and help were indispensable in completing this thesis. First, I would like to thank my advisors, Dr. Thad Starner, Dr. Irfan Essa, and Dr. Jim Rehg, for their thorough guidance of my research and many aspects of my life as a foreign student in this country. The thoughtful discussion about attitudes toward life, a career path, and so forth, always reminded me that you were not only my mentors but also my friends. I would like to thank Dr. Harley Hamilton and Dr. Stan Sclaroff for their participation on my dissertation committee and their valuable feedback to this dissertation. I thank Dr. Aaron Bobick and Dr. Charles Isbell for their insightful critiques while they served as members of my Ph.D. qualifying exam committee. Their challenging questions contributed to the success of my research projects. I also thank other faculty members of the College of Computing, especially those in the School of Interactive Computing, for their knowledgeable input and support in my pursuit of a doctorate degree. The discussion with Dr. Biing-Hwang Juang from the School of Electrical and Computer Engineering and Dr. Vladimir Koltchinskii from the School of Mathematics have also been very productive.

I appreciate the friendship of the students of the Computational Perception Lab, the Wall Lab, the Contextual Computing Group, and the BORG lab. When I become lonely, unhappy, anxious, or frustrated, you are always there for me. I am lucky to have so many friends at Georgia Tech.

I would like to thank the Media Computing group, the Communication Collaboration Systems Group, and the Machine Learning and Perception Group of Microsoft Research, especially my managers, Dr. Xian-Sheng Hua, Dr. Hong-Jiang Zhang, Dr. Harry Shum, Dr. Yong Rui, Dr. Ross Cutler, Dr. Antonio Criminisi, and Dr. John Winn, for offering me interesting, valuable internship opportunities. The experience in the industrial research labs has broadened my prospective and helped to develop my research skills. Many friends in Microsoft have also provided tremendous help in preparation for my oral defense and

finalizing my dissertation.

Finally, I would like to thank Mr. Jim Lovell, Jr. commander of Apollo 13, for his heart-touching speech about faith and perseverance at Georgia Tech in 2007. In the darkest days of my research, I kept remembering the guy sitting in a spaceship with a dead engine, hundreds of thousands of miles from planet earth, never giving up...

I have spent the most precious years of my life at Georgia Tech for this doctorate degree. Everything I have learned will be of a great value in my entire life.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	xi
SUMMARY	xii
I INTRODUCTION	1
1.1 Thesis Statement	1
1.2 American Sign Language Recognition	1
1.3 Discriminative Feature Selection	2
1.4 Basic Unit of Modeling	3
1.5 ASL Verification	4
1.6 Organization	6
II RELATED WORK	8
2.1 Sequence Classification and Sign Language Recognition	8
2.2 Feature Selection for Sequence Classification	10
2.3 Temporal Pattern Discovery	12
2.4 Speech and Sign Verification	14
III SEGMENTALLY BOOSTED HIDDEN MARKOV MODELS	16
3.1 Segmental Boosting	16
3.2 Construction of New Features	18
3.2.1 Data Aggregation by AdaBoost	19
3.2.2 Extension to Multiple Classes	20
3.2.3 The Discriminative New Feature Space	21
IV DISCRIMINATIVE STATE-SPACE CLUSTERING	24
4.1 ASL Phonemes	24
4.2 Inseparable States as the Sub-sign Units	26
4.3 The State-Tying Paradigm	27

4.4	The Iterative Clustering Algorithm	28
V	DISCRIMINATIVE ANALYSIS FOR SIGN VERIFICATION	30
5.1	The CopyCat Project	30
5.1.1	CopyCat, the Game	30
5.1.2	Data Collection	31
5.1.3	Features	34
5.1.4	The CopyCat Dataset	35
5.1.5	The CopyCat-Adult Dataset	36
5.2	Sign Verification	36
5.2.1	Recognizer versus Verifier	40
5.3	Phrase Selection for Game Re-design	40
5.4	Phrase Selection	41
5.4.1	Selection by Classification Accuracy	42
5.4.2	Bi-gram Predictor for Phrase Accuracy	43
5.4.3	Uni-gram Predictor for Phrase Accuracy	44
5.4.4	Measurement of the Quality of Prediction	44
5.4.5	Selection for Unseen Phrases	45
VI	EXPERIMENTAL VALIDATIONS	48
6.1	The Synthetic Experiment	48
6.2	American Sign Language Recognition Results	51
6.3	Human Gait Identification Results	53
6.4	Audio and Visual Speech Recognition Results	55
6.5	Results with Discriminative State-Space Clustering	59
6.6	Probes	62
6.7	Phrase Selection	64
6.7.1	Discussion	74
6.7.2	Bi-gram versus Uni-gram	74
6.7.3	Suggesting New Phrases	75
VII	DISCUSSION AND FUTURE WORK	80
VIII	CONCLUSION	83

APPENDIX A	BASICS OF LINGUISTICS	84
APPENDIX B	STATISTIC TESTS USED IN THIS DISSERTATION	86
REFERENCES	89

LIST OF TABLES

1	The reduction of error by the SBHMM compared to the HMM baseline . . .	17
2	Aggregation by AdaBoost	23
3	An example of a sequential contrast.	25
4	Articulatory linguistic features	26
5	The vocabulary currently used in CopyCat	31
6	The three-sign phrases currently used in CopyCat	32
7	The four-sign phrases currently used in CopyCat	32
8	The five-sign phrases currently used in CopyCat.	33
9	Accuracy metric achieved by the human verifier in CopyCat	34
10	Variations of the signs in the CopyCat dataset	37
11	True positive, false positive, true negative, false negative, and accuracy rates are not independent for ASL verification.	38
12	The four phrases for a simplified game, mini-CopyCat, ranked by an artificial testing accuracy	41
13	All eight possible phrases for mini-CopyCat ranked by their artificial testing accuracy	42
14	Artificial likelihoods for the preposition signs in the five sessions of mini- CopyCat	43
15	Synthetic example (clean)	49
16	Synthetic example (segmental)	50
17	The meaning of the 17 accelerometer readings	53
18	Test error in sign language recognition	53
19	Test error in gait identification	55
20	Georgia Tech Speech Reading Database	56
21	Test error in lip reading and speech recognition	56
22	The two highest generalized Rayleigh quotients	56
23	Average likelihood ratio of the correct over maximal incorrect HMM decoding	56
24	Average variance of the diagonal Gaussians of the HMM	57
25	Top ten confusable states computed by SBHMMs on the third fold of the accelerometer-based ASLR dataset	61
26	Results of DISC-SBHMMs	61

27	Paired t-test of HMMs, SBHMMs, and DISC-SBHMMs	61
28	Top six confusable state clusters (from top ten pairs) computed by SBHMMs for one subject of the CopyCat dataset.	62
29	CopyCat verification results with manually specified thresholds and the au- tomatically computed threshold by the probe	65
30	CopyCat-Adult verification results with manually specified thresholds and the automatically computed threshold by the probe	65
31	The training error ranking of the CopyCat data	67
32	The training error ranking of the CopyCat-Adult data	68
33	The true error ranking of the CopyCat data	70
34	The true error ranking of the CopyCat-Adult data	72
35	Spearman footrule distance computed from the CopyCat data.	73
36	Spearman footrule distance computed from the CopyCat-Adult data.	73
37	Kendall tau distance and Pearson product-moment correlation coefficient computed from the CopyCat data.	74
38	Kendall tau distance and Pearson product-moment correlation coefficient computed from the CopyCat-Adult data.	74
39	Comparison of uni-gram versus bi-gram models using CopyCat data	75
40	Comparison of uni-gram versus bi-gram models using CopyCat-Adult data .	75
41	Result of leave-one-out (L1O) test compared with others using CopyCat data	76
42	Twenty three-sign phrases for CopyCat suggested by this dissertation . . .	76
43	Twenty four-sign phrases for CopyCat suggested by this dissertation	77
44	Nineteen five-sign phrases for CopyCat suggested by this dissertation	77
45	Twenty three-sign phrases for CopyCat-Adult suggested by this dissertation	78
46	Twenty four-sign phrases for CopyCat-Adult suggested by this dissertation	78
47	Nineteen five-sign phrases for CopyCat-Adult suggested by this dissertation	79

LIST OF FIGURES

1	A margin distribution graph showing the increase in the minimum margin and the decrease in the average margin	20
2	Splitting of the HMM states	27
3	The CopyCat game	30
4	The game kiosk	34
5	Data gloves used in CopyCat	35
6	Visual features for CopyCat	35
7	Some signing variations	36
8	K-shortest path	47
9	The impact of SBHMMs on synthetic data	51
10	HMMs used in sign language recognition	52
11	Illustration of the formation of the minimal pair BROTHER and SISTER .	53
12	The impact of minimal pairs in feature weighting obtained by the SBHMMs	54
13	The impact of SBHMMs on lip reading data	57
14	Statistical measurements	60
15	ROC curve of the verification results in CopyCat.	65
16	ROC curve of the verification results in CopyCat-Adult.	66
17	Ranking approximation measured by Spearman footrule distance in the Copy-Cat data.	71
18	Ranking approximation measured by Spearman footrule distance in the CopyCat-Adult data.	73
19	The new handshape features	81

SUMMARY

This dissertation presents segmental discriminative analysis techniques for American Sign Language (ASL) recognition and verification. ASL recognition is a sequence classification problem. One of the most successful techniques for recognizing ASL is the hidden Markov model (HMM) and its variants. This dissertation addresses two problems in sign recognition by HMMs. The first is discriminative feature selection for temporally-correlated data. Temporal correlation in sequences often causes difficulties in feature selection. To mitigate this problem, this dissertation proposes segmentally-boosted HMMs (SBHMMs), which construct the state-optimized features in a segmental and discriminative manner. The second problem is the decomposition of ASL signs for efficient and accurate recognition. For this problem, this dissertation proposes discriminative state-space clustering (DISC), a data-driven method of automatically extracting sub-sign units by state-tying from the results of feature selection. DISC and SBHMMs can jointly search for discriminative feature sets and representation units of ASL recognition.

ASL verification, which determines whether an input signing sequence matches a pre-defined phrase, shares similarities with ASL recognition, but it has more prior knowledge and a higher expectation of accuracy. Therefore, ASL verification requires additional discriminative analysis not only in utilizing prior knowledge but also in actively selecting a set of phrases that have a high expectation of verification accuracy in the service of improving the experience of users. This dissertation describes ASL verification using CopyCat, an ASL game that helps deaf children acquire language abilities at an early age. It then presents the “probe” technique which automatically searches for an optimal threshold for verification using prior knowledge and BIG, a bi-gram error-ranking predictor which efficiently selects/creates phrases that, based on the previous performance of existing verification systems, should have high verification accuracy.

This work demonstrates the utility of the described technologies in a series of experiments. SBHMMs are validated in ASL phrase recognition as well as various other applications such as lip reading and speech recognition. DISC-SBHMMs consistently produce fewer errors than traditional HMMs and SBHMMs in recognizing ASL phrases using an instrumented glove. Probe achieves verification efficacy comparable to the optimum obtained from manually exhaustive search. Finally, when verifying phrases in CopyCat, BIG predicts which CopyCat phrases, even unseen in training, will have the best verification accuracy with results comparable to much more computationally intensive methods.

CHAPTER I

INTRODUCTION

1.1 Thesis Statement

This dissertation presents data-driven segmental discriminative analysis techniques for American Sign Language recognition/verification and other sequence classification tasks. These techniques, segmentally-boosted hidden Markov models (SBHMMs), discriminative state-space clustering (DISC), and bi-gram error-ranking (BIG) prediction can perform the following tasks:

- extract discriminative features to improve recognition accuracy,
- extract and share common “sub-sequence units” to reduce recognition complexity,
- generate the “most distinguishable” phrases for verification tasks.

The meaning, the novelty, the importance and the evaluation of these capabilities are explained in the remainder of this chapter.

1.2 American Sign Language Recognition

The natural language for most deaf signers in the United States is American Sign Language (ASL). ASL sentences are composed of signs, such as UNCLE, EAT, and SAD, with a quite different grammar than English. In machine perception, American Sign Language recognition (ASLR) algorithms infer sign words from sensor readings such as a video stream of the signer’s hand movements. Ong and Ranganath [73] provide a detailed review of the technologies used in ASLR. Most recent studies on ASLR have moved from isolated word-level ASLR to continuous sentence-level ASLR, which is more practical. Continuous ASLR contains (at least) two levels of recognition. While sentence-level recognition demands a sequential recognition of each individual sign, word-level recognition in continuous ASLR identifies a single sign label from a sub-sequence just like isolated ASLR, but with unknown

boundaries in a continuous stream of signs. This dissertation focuses on improving the accuracy of word-level recognition in continuous ASLR. As defined by Dietterich [20]: a sequential classification problem predicts a series of labels from a series of observations, while a sequence classification problem predicts a single label from an entire observation sequence. Therefore, the purpose of this dissertation is to analyze how the accuracy of sequence classification can be improved.

Since Stokoe demonstrated that ASL, like spoken languages, is compositional with an internal structure [94], various signal processing and machine learning techniques successful at speech recognition have been applied to ASLR for higher accuracy and efficiency. Despite the strong similarities between spoken and sign languages, several ASL-specific issues must be addressed.

1.3 Discriminative Feature Selection

The first ASL-specific issue pertains to the input features¹ used in ASLR. Linguists use articulatory (linguistic) features to form bundles called phonemes and study the models of how these phonemes compose words and signs. Proposed by Liddell and Johnson [56], the movement-hold model describes ASL by two types of sequentially-ordered segments: movement segments (M) and hold segments (H). In the movement-hold model, linguistic features, such as handshape, are organized in sequential segments. The movement-hold model justifies the application of hidden Markov models (HMM) [80], commonly used in speech recognition, to American Sign Language recognition (ASLR). In fact, most research groups are using HMMs for sign language recognition.

While the movement-hold model provides a powerful linguistic tool with which humans can analyze ASL, the “conceptual” descriptors such as hand posture used by the movement-hold model may not be available to machines. For automatic ASLR by machines, the inputs are usually pixel values in vision-based systems or sensor readings in glove-based systems. One may suggest a two-step approach that first recognizes all conceptual descriptors and

¹That is, the sensor reading and/or other sources of input that compose the input vector for recognition should not be confused with the linguistic term “feature” listed in Appendix A. To avoid such confusion, this dissertation uses the term “feature” to refer to the input for machine perception and the term “linguistic feature” to refer to the linguistic descriptor.

then applies the phonological rules. However, the variance of signing, disfluencies [84], inaccurate recognition results, and many other factors cause inferior performance in practice. Alternatively, ASLR research has proven that low-level features can fit the phonemic models of ASL such as the movement-hold model directly [98].

Informative features improve recognition efficiency and generalization, just as the introduction of the Mel-frequency cepstral coefficients (MFCCs) [80] contributed to the success of speech recognition. Although some linguistic studies suggested several (linguistic) “distinctive features” [50, 83, 25] for human perception, computer scientists do not yet have a set of “good” features for machine perception. The manually-designed features for ASL, usually the readings of some trackers or sensors selected based on human knowledge, may not be the most useful for machine recognition. However, automatic feature selection algorithms, which usually assume that the data are independent and identically distributed, cannot directly be applied to ASL sequences, which have a strong temporal correlation and variation. This dissertation proposes a segmental feature extraction algorithm, segmentally-boosted hidden Markov models (SBHMMs), that accounts for temporal dependency and automatically constructs a discriminative feature space in which hidden Markov models with Gaussian observations achieve higher recognition accuracy.

1.4 Basic Unit of Modeling

The second ASL-specific issue pertains to the appropriate building blocks for ASL. In linguistics studies, the building blocks used are phonemes, morphemes, syllables, and so on (see Appendix A for details). These building blocks are necessary for scalability in large vocabulary ASLR as they are in speech recognition. Following the conventions of phonology in spoken languages, the sub-sign building blocks in ASL linguistics are called “phonemes” (or “cheremes” by Stokoe [94]). Phonemes, the smallest contrastive units of a language, can be illustrated in minimal pairs. Minimal pairs in sign languages are two different signs that are identical except for one of the three major formational categories (hand shape, location and movement). Depending on the types of contrasts exhibited, sign language phonologists have proposed different phonemic models, such as the Stokoe system [94] and the

movement-hold model [55]. Previous research on ASLR has “not yet exploited the results of determining the appropriate basic units” [35]. This dissertation, instead of investigating an appropriate set of “ASL phonemes,” will focus on extracting data-driven sub-sign units to improve recognition accuracy by machines.

In addition to the scalability issue mentioned before, extraction of the correct basic units affects the granularity of feature selection, which accounts for the success of recognition. Supervised feature selection produces features that are “optimal” for only specified classification tasks such as phonemes or signs, and the selected features determine the performance of recognition. Therefore, this research proposes a data-driven sub-sign unit extraction algorithm, called discriminative state-space clustering (DISC), in conjunction with the segmental feature selection algorithm described above. The intuition is that the inseparable clusters found by discriminative feature selection are likely to be a basic “building block” shared by sequences (signs).

1.5 ASL Verification

Early exposure to language is important to children’s development of language ability and short-term memory. However, 95% of deaf children are born to hearing parents [65]. Because most of these parents are unable to sign fluently, the first exposure of fluent sign language to their children is often delayed until the deaf children are school age. Therefore, the majority of deaf children of hearing parents, who remain significantly delayed in language development throughout their lives when compared with hearing children and deaf children of deaf parents [40, 91, 92], can be considered semilingual [31, 44] as they are fluent in neither English nor ASL. For these deaf individuals, semilingualism is sometimes a life-long struggle [5].

In order to assist young deaf children with early language acquisition, interactive sign language games, such as CopyCat [54] shown in Figure 3 in Chapter 5, have been developed. CopyCat allows deaf children, who wear colored gloves and wrist-mounted accelerometers, to communicate with an animated hero, Iris the Cat, in a computer game using ASL. In CopyCat, the children interact with the hero via sign language in a talking scene to inform

the hero where a “bad” animal guard is hiding. If the signing is verified by the ASLR system as correct, the hero will “poof” the guard and retrieve a desired item; otherwise, the hero will look confused and encourage the user to sign again.

The goal of the game is not to teach the complete vocabulary of ASL but to initialize deaf children’s language ability and build their short-term memory. To achieve this goal, the game emphasizes user experience because children are less patient with “flawful interaction” than adults. Thus, two factors that improve user experience should be taken into account: (1) given a game design, how can modification of the ASLR algorithm reduce error (addressed in the data-driven segmental discriminative analysis in Chapters 3 and 4) and (2) given an imperfect ASLR algorithm, how can modification of the game reduce error (addressed in Chapter 5).

For the second issue, this dissertation introduces a new concept for *ASL verification*². Compared with traditional ASL recognition, ASL verification has two additional challenges. First, ASL recognition classifies a valid signing sequence into one of the known sign labels, and this process can be evaluated by a single metric of recognition accuracy. ASL verification in this dissertation, by contrast, determines whether a given sample of signing and a label match. Such a task requires five metrics: the true positive rate, the false positive rate, the true negative rate, the false negative rate, and accuracy; however, these metrics are not independent, as shown in Table 11 in Chapter 5. In ASL verification, while additional information (that is, the label) is available as prior knowledge, a naïve algorithm based on similarity can easily produce trivial results, as described in Chapter 5. Therefore, the first challenge is to find an optimal threshold for the similarity measurement of match versus no-match.

The second challenge is related to the choice of the verification content: the analysis of results for the re-design of the game. The verification process, including both the design

²This is slightly different than the definition by Yang, *et al.* [106]. Their verification is to decide whether to accept a spotted sign using additional appearance information. In this dissertation, the “groundtruth” script of signing is already known to the verification system. An input that deviates from this script, such as a combination of correct signings and out-of-vocabulary movements, will be rejected in our educational application; that is, we are literally “verifying” a script. More discussion comparing the two types of verification is available in Chapter 2.

of the verification phrases and the recognizer, is optimized for high recognition accuracy for a pre-defined script of phrases. This pre-selection of highly distinct phrases is different than the regular recognition process, which is optimized for overall recognition accuracy for a much less constrained utterance. In this sense (the “groundtruth phrases” are known), ASL verification is similar to speech verification [52], where the content of an utterance is verified by a speech recognizer as a proof of identity. Speech systems provide several examples of vocabulary chosen for its distinctiveness. The NATO phonetic alphabet [1], which uses ‘Alpha’ to replace “A,” “Bravo” to replace “B,” *etc.* , selects vocabulary to reduce confusion in verification; another example is in military communications, where “roger” and “affirmative” are used instead of the words “OK” or “yes” to confirm orders. Similarly, an objective for studying sign verification in CopyCat is to design ASL phrases that are easy to verify so that children will not get frustrated by verification errors made by the computer. A straightforward strategy for designing such phrases is to train a standard HMM-based ASLR system, compute the confusion matrix of the ASL signs, and output a ranked list of the least confusing combination of ASL phrases. However, enumerating all possible combinations of phrases that can be used in the game is computationally intractable. This dissertation proposes building a “quality” predictor, called the bi-gram error-ranking (BIG) predictor, for signing phrases based on partial, segmental information of other phrases. Such a predictor will be able to determine in polynomial time whether an unseen phrase should be included in the new game design to improve verification accuracy. We believe that the success of a phrase quality prediction algorithm for sign verification could be extended to other domains, such as speech verification, voice prompting, and so on.

1.6 Organization

The next chapter reviews the challenges and related studies of feature selection for sequence classification, sequential pattern discovery, and verification. Chapter 3 proposes segmentally-boosted hidden Markov models (SBHMMs), a discriminative feature selection

algorithm for ASLR using a rough approximation in the state space. Chapter 4 proposes discriminative state-space clustering (DISC), a data-driven sub-sign unit extraction algorithm that works jointly with SBHMMs for ASL. Chapter 5 describes a discriminative analysis of sign verification and phrase selection in the application of the Georgia Tech CopyCat project. Chapter 6 describes the evaluation methodology for DISC-SBHMMs and BIG and the experimental results. Chapter 7 discusses the results from the experimental studies and suggests future work, Chapter 8 concludes with a summary of the contributions of the research.

CHAPTER II

RELATED WORK

The first part of this chapter reviews related work in sequence classification, specifically sign language¹ recognition, followed by feature selection and pattern discovery techniques used for sequence classification. The second part of this chapter reviews works related to sign verification.

2.1 Sequence Classification and Sign Language Recognition

Research on sign language recognition started with isolated signs in the 1990s [48, 99]. Nowadays, research groups have turned their attention to a more difficult, practical recognition task, continuous ASLR [93, 28, 98]. Vocabulary sizes range from five [12] to 5,113 [15] different signs. Loeding *et al.* [57] provide a detailed review of the recent advances in sign language recognition.

Sign language recognition, as well as speech recognition, gesture recognition, DNA analysis, and so on, is a sequence classification task. One major difference between sequence classification and “static” classification tasks (*e.g.* face recognition) is that the input signal of the former has a variable length. In order to compare such sequences, we must observe some form of temporal (length) invariance. One way to achieve temporal invariance is spectrum analysis (*e.g.* Fourier analysis), which converts a signal from the time domain to the frequency domain. The other way to achieve invariance is temporal alignment. An example of a non-parametric temporal alignment technique is dynamic time warping (DTW) [80], and examples of parametric temporal alignment techniques include switching linear dynamic systems (SLDS) [75], hidden Markov models (HMMs) [80], and conditional random fields (CRFs) [49]. Both DTW and SLDS have a continuous-time space, while both HMMs and CRFs have a discrete-time space. Because ASL has sub-sign building blocks [94], this

¹Not just ASL.

dissertation limits the discussion to discrete-time space models.

The HMM and its variants, such as parallel-HMMs [98], which assume that the input signal is generated by a latent discrete-time Markov process, are arguably the most successful models for sign language and speech recognition. A detailed review of HMMs can be found in Rabiner and Juang [80]. One questionable assumption made by HMMs is that all observations are conditionally independent. Semi-Markov models, such as hidden semi-Markov models (HSMM) [69] and semi-Markov conditional random fields (semi-CRF) [85], assume a Markovian property only between temporal segments and allow non-Markovian behavior inside those segments. Therefore, features at the segment level such as the length (duration) of a segment can be used in semi-Markov models. Semi-Markov models, while more expressive, significantly increase computational and numerical complexity [81]. This dissertation further limits the discussion to the processes that can be modeled by HMMs and explicitly re-uses the conditional independence assumption for discriminative feature selection.

An HMM is normally estimated by maximum likelihood estimation (MLE). This generative method requires model correctness and infinite training data to be optimal. In practice, however, MLE is not as effective for classification as discriminative methods, which simply learn a decision boundary. Previous attempts to introduce discriminative methods to HMMs can be classified into two categories: discriminative training of model parameters and discriminative feature selection. Discriminative variants of HMM parameter training, *e.g.*, minimum classification error (MCE) [41], maximum mutual information (MMI) [80] and conditional maximum likelihood (CML) [104] criteria directly adjust model parameters for classification. A detailed review and comparison of such techniques can be found in Sha and Saul [88]. In addition to these discriminatively trained models, previous studies [34, 62, 19] indicate that discriminative features facilitate more accurate and efficient recognition by emphasizing informative features and filtering out irrelevant ones. Therefore, selecting/extracting discriminative features for HMMs, a focus of attention [18, 77, 110, 58], represents the category to which our segmentally-boosted HMM (SBHMM) technique belongs. We focus on extracting discriminative features for classification with an explicit

consideration of temporal correlation. We review feature selection algorithms next.

2.2 Feature Selection for Sequence Classification

In machine learning, automatic feature selection is usually cast as the optimization of a supervised classification problem. If x is evidence, and y is a label, the pair (x, y) defines a classification problem $y = f(x)$, and the discriminative features are computed in order to minimize classification loss. However, the application of such feature selection methods for time sequences results in two major difficulties.

One is that sequential data do not obey the basic assumption of supervised learning, *i.e.*, that the samples are independent and identically distributed (i.i.d.). Time sequences contain a significant amount of temporal correlations (not independent sampling); some sequences may also contain several phases/states, in which the discriminative features for one phase may be quite uninformative for another (not identically distributed). For instance, the sign FISH in ASL is expressed by moving the two hands asynchronously.

If the state of a sequence can be clearly identified, for example, in part-of-speech tagging [49] and video layer segmentation [109], conditional models such as CRFs [49, 106] and large margin HMMs [89] can be applied to perform sequential classification [20] (to predict a sequence of labels for a sequence of observations). However, the meaning and the labeling of the states are mostly unavailable in sequence classification [20] (to predict a single label for an *entire* sequence). For example, to recognize the sign BROTHER, how can a human labeler precisely supervise the training for the first state when he/she does not even know (1) the meaning of this state or (2) how many states compose the sign? In such situations, SBHMMs provide a solution to obtain similar discriminative classifiers in an unsupervised manner. As we mention later, SBHMMs can be modified to refine iteratively the selected features jointly with the Baum-Welch re-estimation.

In order to account for the unknown temporal structure while maintaining the discriminative ability, hidden conditional random fields (HCRF) [78] introduce hidden state variables to CRF. In HCRF, hidden states and observation sequences form a CRF, except that the exact state assignments are unknown. The states are further modulated by

one sequence label which is observable. On the one hand, HCRF generalizes HMMs, because it can simulate an HMM by setting the feature functions appropriately; on the other hand, it generalizes CRFs because a CRF is one HCRF whose state assignments are known and fixed. HCRFs have been successfully applied to phone classification (speech) [30] and gesture recognition [78]. HCRFs can only model segmented sequences, which requires a pre-processing step for continuous input. In order to model both intrinsic structure and inter-label interaction, LDCRFs [66] introduce both a hidden layer and a continuous stream of labels for an input sequence. In order to stay computationally tractable, LDCRFs require that different classes having a disjoint set of the hidden states. LDCRFs have been validated in binary sequential classification problems such as head motion and eye gaze datasets with manual labeling. The difference between CRFs (including CRF/HCRF/LDCRF) and the proposed SBHMMs, in CRF terminology, is that CRFs discriminatively compute the weight for a set of manually-defined feature functions while SBHMMs *automatically select* from a family of feature functions and discriminatively compute their weights. However, while SBHMMs achieve flexibility in feature selection at the cost of breaking feature selection and temporal recognition into two steps, CRFs can optimize the entire recognition process in a unified framework by L-BFGS [87] or stochastic gradient descent (SGD). Thus, CRFs are effective in tuning feature weights for applications in which a set of good features is already known; and SBHMMs are effective for applications in which feature selection is necessary. In addition, embedding training [80, 103] of SBHMMs/HMMs allows labeled, but unsegmented, input sequences for training.

Another problem with the application of the feature selection method is that sequences are variable in length even though the learning functions $f(\cdot)$ usually expect inputs x of fixed dimensionality. In order to handle this dilemma, the Fisher kernel [36] and its variants use a generative model to preprocess the sequences and construct a discriminative kernel according to the Fisher score (local gradient) of that generative model. The kernel contains “feature weighting” according to their discriminative ability. The Fisher kernel methods have been successfully applied to domains such as bioinformatics. However, its reliance on a potentially imperfect generative model can cause problems in the initialization of the

discriminative learning.

The variable length problem can also be solved by temporal alignment. Lv *et al.* [58] constructed multi-HMM classifiers for activity recognition data. Each of the HMMs contains a set of observation features that correspond to the motion of a body part, and boosting assigned weights to those HMMs according to their discriminative ability. This idea is similar to the “boosting multi-HMMs” approach in speech recognition [63]. The design of the multi-HMMs requires prior knowledge about the grouping of the features. Furthermore, that boosting assigns weights to the entire HMMs precludes the possibility of capturing segmental features that are discriminative only in part of the sequences. Other segmental models, such as semi-CRF [85] and segmental SLDS [72], either are very expensive to compute or assume a continuous time space.

Recently, feature-space minimum phone error (fMPE) [77] and stereo-based piecewise linear compensation for environments (SPLICE) [18] have produced significant improvements in large vocabulary recognition accuracy, for they adjust input features with posterior-based “correction vectors” [19]. However, they are relatively expensive to compute in practice.

2.3 Temporal Pattern Discovery

ASL signs can be decomposed to sub-sign parts [94], and sign language experts have designed annotations for ASL phonemes [97]. A belief held in this dissertation is that statistical pattern discovery (discriminative state-tying) may yield a set of data-driven units for ASLR for better recognition. If these units are directly optimized for completeness and compactness in representation, we expect a recognition model with reduced complexity while still maintaining comparable, if not higher, accuracy to systems without such abstraction. This section reviews the literature on data-driven pattern discovery.

In “static” pattern discovery, the most straightforward method is unsupervised clustering, such as k-means, iterative conditional modes (ICM), and expectation maximization (EM) algorithms [9]. Cluster assignments are then used as a codebook for computing shared parameters [80] or extracting features for further processing [108]. Such techniques

are widely used in speech recognition [79, 59], music analysis [17, 38], computer vision [43], and so on.

In temporal pattern discovery, the speech community has long been using vector quantization (VQ) [59] to build codebooks for more efficient representation and processing. Keogh *et al.* [46] proved that simply applying static clustering methods to sub-sequences can easily lead to trivial results. In Chiu *et al.* [16], probabilistic motif extraction was proposed to ensure the quality of temporal pattern discovery. Recently, Minnen *et al.* [64] have developed a sub-dimensional multivariate pattern discovery strategy. Hamid *et al.* [32] applied a suffix tree to activity analysis and anomaly detection. These motif discovery techniques focus on representation and detection problems in symbolic sequences, while our approach is to reduce classification error in a data stream of sensor readings.

To achieve scalability in sign language recognition, phonemic models in linguistics have been applied to model temporal patterns in signing. For example, Vogler and Metaxas [98] used parallel-HMMs with the Bakis topology [80] to simulate an extended movement-hold model for about 40 ASL phonemes. Wang *et al.* [100] learned about 2,400 HMMs for phonemes of Chinese Sign Language. In both studies, the linguistic phonemes are also manually transcribed and designed specifically for HMMs for scalability in recognition. However, such a transcription process is manually intensive for a large vocabulary size. To acquire statistically meaningful temporal units automatically, researchers in speech recognition proposed the concept of “fenones” [6] (or acoustic segment units, ASU, in Rabiner and Juang [80]), in contrast to “phonemes.” While phonemes are manually defined for linguistic purposes, the fenones are completely data-driven and optimized for recognition using k-means clustering. In ASLR research, Bauer and Kraiss [7] applied a similar idea to extracting ASL fenones. Han *et al.* [33] used motion speed discontinuity and motion trajectory discontinuity to cluster segments of signs and extract visually meaningful sub-sign units. However, such sub-sign clustering methods reduce only the encoding complexity without explicit optimization for discriminative features, as opposed to the DISC-SBHMMs proposed in this dissertation. Nayak *et al.* [70, 71] used DTW to extract signemes in multiple sentences, and computed the start and end frame of the signemes by iterative conditional

modes (ICM) [8]. Signemes, invariant to co-articulation, serve as the “core parts” of signings, and they can be used for spotting. This technique assumes that a signeme to be extracted is the only common sub-sequence in a collection of training sentences. Signeme extraction may be extended to sub-sign unit extraction (1) if such units are known and (2) a signeme is the only common component of a set of training sentences. In contrast, our DISC technique is designed to discover *unknown* sub-sign units shared in *some* signs by discriminative analysis with the purpose of maintaining accuracy in a reduced-complexity model.

2.4 Speech and Sign Verification

Sign verification has been used in validating sign spotting [3, 106], that is, *pruning the results* of the initial spotting algorithm. In this dissertation, we focus on *comparing groundtruth with inputs*. The two types of verification differ in the following ways. First, Alon and Yang [3, 106] verify a prediction, while this dissertation verifies a known groundtruth script. Second, Alon and Yang [3, 106] identify signs from a vocabulary in an input stream that contains garbage classes, while this dissertation rejects an input stream with disfluencies since the application is for educational purposes.

The closest work to the sign verification defined in this dissertation is speech verification, specifically, as used for computer-aided language learning (CALL) [24]. Early studies such as project Fluency [27] directly use speech recognition for tutoring the pronunciation of foreign language learners. However, “if a tutor explicitly announces whether a student read a word correctly, or colors each word red that it thinks the student misread, then ASR errors place its credibility at risk” [67]. In project “Literacy Innovation that Speech Technology ENables” (LISTEN) at CMU, which displays a story on the screen and “uses speech recognition to listen to children read aloud” [68], Mostow described a motivational cost as the penalty for false negatives and a cognitive cost as the penalty for false positives and proposed “implicit judgments about individual words.” He also discussed metrics to access the recognition accuracy for the speech verification task.

In sign language tutoring, DeSIGN [105] adapts the student model used in Project

LISTEN with the goal of “reinforcing the meaning and use of vocabulary” of ASL for deaf children. In contrast, the primary goal of CopyCat is the development of language ability, not vocabulary.

Brashear [13] proposes to investigate and characterize disfluent signing such as scratching, fidgeting, false starts, hesitations, and pauses to aid in the pattern recognition task. Researchers have devoted extensive efforts to detecting disfluent reading/signings for verification [114, 67, 10, 13], because (1) disfluency detection improves accuracy [67], and (2) disfluency, which at least correlates to the perceptual quality of speech, provides an additional means for assessment [10]. However, to the best of our knowledge, this dissertation is the first to suggest computationally the content of the verification task to improve user experience.

CHAPTER III

SEGMENTALLY BOOSTED HIDDEN MARKOV MODELS

This dissertation proposes SBHMMs [111], which leverage both the dynamic nature of sequential data and the static nature of large-margin feature selection methods by assuming the data are *piecewise independent, identically distributed (piecewise i.i.d.)*. Note that this assumption does not introduce additional approximations because it is already assumed by HMMs. Our experiments show that SBHMMs reduce the sequence recognition error of HMMs by 17% to 70% in American Sign Language recognition, human gait identification, lip reading, and speech recognition (see Table 1, with details in Chapter 6). SBHMMs construct new features by comparing the feature value with a set of discriminatively chosen thresholds, which can be efficiently computed. The key steps to our SBHMM technique are illustrated below.

Algorithm 1 The Segmentally-Boosted Hidden Markov Models (SBHMMs) Algorithm

- 1: Train HMMs by Expectation Maximization (EM) using the time sequence training data.
 - 2: Find the optimal state transition path by the Viterbi decoding algorithm.
 - 3: Label every sample with its most likely hidden state.
 - 4: Train AdaBoost ensembles for this labeling.
 - 5: Project the data to a new feature space using the ensembles.
 - 6: Train HMMs by EM in the new feature space.
 - 7: In testing, project the test data to the same new feature space and predict their label using the HMMs computed in Step 6.
-

3.1 Segmental Boosting

In the 1990s, Juang and Rabiner [42] introduced the idea of segmental training, which computes a better initial estimation for the observation models using the data segments obtained from the Viterbi algorithm [79]. The assumption is that the initial state assignment by the Viterbi algorithm roughly represents the characteristics of the data. In this work, we extend the concept of segmental training in order to perform discriminative feature selection. We derive this strategy in the context of the first-order HMM, which has been

very successful in interpreting temporal data.

Table 1: The reduction in error by the SBHMM compared to the HMM baseline in the five experiments conducted in Chapter 6.

ASLR (vision)	ASLR (accelerometer)	Gait Recognition	Lip Reading	Speech Recognition
36.4%+	17.1%+	70.1%	32.2%	39.2%

An HMM builds a causal model for an observation sequence $\mathbf{O} = (o_1 o_2 \cdots o_T)$ by introducing corresponding “hidden states” $\mathbf{q} = (q_1 q_2 \cdots q_T)$. Let $P(q_1) = P(q_1|q_0)$. The transition model is $P(q_t|q_{t-1})$, and the observation model is $P(o_t|q_t)$. Assuming the presence of C types of sequences, recognition selects the one with the highest likelihood

$$c^* = \arg \max_{1 \leq c \leq C} P(\mathbf{O}|\lambda_c), \quad (1)$$

and $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_C\}$ are the parameters of the HMMs.

For a type c sequence \mathbf{O}^c with length T_c , we define the model distance (dissimilarity) [80] as

$$D(\lambda_c, \Lambda) = \frac{1}{T_c} [\log P(\mathbf{O}^c|\lambda_c) - \frac{1}{C-1} \sum_{v \neq c} \log P(\mathbf{O}^c|\lambda_v)]. \quad (2)$$

We intend to choose a subset of features that maximize $D(\lambda_c, \Lambda)$. Assuming an uninformative prior, it is equivalent to maximizing the “sequence margin”

$$M(\lambda_c, \Lambda) = \frac{1}{T_c} [\log P(\lambda_c|\mathbf{O}^c) - \frac{1}{C-1} \sum_{v \neq c} \log P(\lambda_v|\mathbf{O}^c)]. \quad (3)$$

Discriminative classifiers with logistic output, such as boosting $H(x) = \sum_j \alpha_j h_j(x) = \log P(y = y^*|x) - \log P(y \neq y^*|x)$ are capable of maximizing such a margin in Equation 3 for the classification problems (x, y) .

The three natural choices for (x, y) represent different granularities: $(x = \mathbf{O}, y = c)$, $(x = o_t, y = c)$, and $(x = o_t, y = q_t)$. The first is intractable since the length of observation \mathbf{O} varies. The second corresponds to the sliding window methods [20] with fixed [110] or empirically determined [90] size. Although improved results are reported, the oversimplified assumption limits its application to more complicated tasks in which “the static features

tend to cluster...without dynamic information” [58]. Therefore, we should not neglect the sequential (temporal) dependency between the sliding windows, which conveys important information for recognition.

We argue that the temporal dependency can be *decoupled* from the discriminative feature selection process instead of being discarded. In order to stay tractable, we assume that the sequence data are piecewise (state-wise) i.i.d. and introduce a set of supervised learning problems ($x = o_t, y = q_t$) to select features. This labeling preserves the temporal relationship between the learning problems while assuming the samples for each problem are least correlated. The assumption of piecewise i.i.d. follows the HMM assumption about the Markov property and conditional independence:

$$\begin{aligned} P(\mathbf{O}|\lambda_c) &= \sum_{\mathbf{q}} P(\mathbf{O}|\mathbf{q}, \lambda_c) P(\mathbf{q}|\lambda_c) \\ &= \sum_{\mathbf{q}} \prod_{t=1}^T P(o_t|q_t) P(q_t|q_{t-1}). \end{aligned}$$

Thus, $M(\lambda_c, \Lambda)$ can be increased with some discriminative $P(o_t|q_t)$. Intuitively, HMMs decompose the evolving temporal trajectory into two types of behavior: (1) a loop within the same state or (2) a transition from one state to another. Thus, we can perform feature selection only in the segments of the same state, and “static” segments are connected by the temporal transition $P(q_t|q_{t-1})$. Note that the concept of “hidden state” is still necessary to smooth the results of the observation model.

Therefore, we first train a set of HMMs with the original features and label every observation o_t by its maximum *a posteriori* (MAP) state s_t computed by the *Viterbi* algorithm. Then we train a set of AdaBoost [29] ensembles $\{H^{(s)}\}$, that consists of decision stumps (thresholding functions), for such labeling. We ignore superscript s when we discuss these ensembles in general.

3.2 Construction of New Features

We use boosting classifiers to compute a new feature space that is discriminative for sequence classification. In this dissertation, new features are constructed in a similar manner as in the tandem model [34]. However, the tandem model is usually for labeled phonemes in speech

recognition while our segmental method does not have access to such labeling. In addition, the construction of our discriminative feature space is based on the margin properties of the AdaBoost algorithm, described as follows.

3.2.1 Data Aggregation by AdaBoost

AdaBoost linearly combines weak learners $h_j(x_i) \in [-1, 1]$ to obtain a strong classifier (ensemble) $H(x) = \sum_j \alpha_j h_j(x)$ for each class s (state). Weak learners h used in SBHMMs are the decision stumps, such as “Is the value of feature No. 5 greater than 0.45?” The binary answers are then weighted according to their empirical discriminative power in separating class s from the other classes.

The margin of the ensemble with l weak learners at x_i is defined as $m_l(x_i) = \frac{y_i H_l(x_i)}{w_l}$, while $w_l = \sum_{j=1}^l \alpha_j$ is the sum of the learner weight, serving as a normalization factor. During AdaBoost training, the minimum margin $\min_i \{m_l(x_i)\}$ tends to increase [86], which leads to good generalization ability.

We show that *the average margin tends to decrease* as the training proceeds. The average margin of AdaBoost at round l and that at round $l+1$ are compared:

$$\begin{aligned} \overline{m}_l &= \frac{\sum_{i=1}^n y_i H_l(x_i)}{n \cdot w_l} = \frac{\sum_{i=1}^n y_i \left[\sum_{j=1}^l \alpha_j h_j(x_i) \right]}{n \cdot \sum_{j=1}^l \alpha_j} = \frac{\sum_{j=1}^l \sum_{i=1}^n \alpha_j h_j(x_i) y_i}{\sum_{j=1}^l \sum_{i=1}^n \alpha_j y_i^2} \\ \overline{m}_{l+1} &= \frac{\sum_{i=1}^n y_i H_{l+1}(x_i)}{n \cdot w_{l+1}} = \frac{\sum_{i=1}^n y_i \left[\sum_{j=1}^{l+1} \alpha_j h_j(x_i) \right]}{n \cdot \sum_{j=1}^{l+1} \alpha_j} = \frac{\sum_{j=1}^{l+1} \sum_{i=1}^n \alpha_j h_j(x_i) y_i}{\sum_{j=1}^{l+1} \sum_{i=1}^n \alpha_j y_i^2} = \frac{\sum_{j=1}^l \sum_{i=1}^n \alpha_j h_j(x_i) y_i + \sum_{i=1}^n \alpha_{l+1} h_{l+1}(x_i) y_i}{\sum_{j=1}^l \sum_{i=1}^n \alpha_j y_i^2 + \sum_{i=1}^n \alpha_{l+1} y_i^2}. \end{aligned}$$

Denote

$$\begin{aligned} A &= \sum_{j=1}^l \sum_{i=1}^n \alpha_j h_j(x_i) y_i, & B &= \sum_{j=1}^l \sum_{i=1}^n \alpha_j y_i^2, \\ C &= \sum_{i=1}^n \alpha_{l+1} h_{l+1}(x_i) y_i, & D &= \sum_{i=1}^n \alpha_{l+1} y_i^2. \end{aligned}$$

We have

$$\overline{m}_l = \frac{A}{B}, \quad \overline{m}_{l+1} = \frac{A+C}{B+D},$$

in which

$$\overline{m_{l+1}} \geq \overline{m_l} \Leftrightarrow \frac{A+C}{B+D} \geq \frac{A}{B} \Leftrightarrow \frac{C}{D} \geq \frac{A}{B}.$$

In practice, “<” happens much more frequently than the other two cases, because (1) A/B is the classification error of the ensemble composed by l weak learners while C/D is the error of one weak learner at $l + 1$; and (2) AdaBoost optimization gradually focuses on the “harder” examples [86]. Therefore, $\overline{m_l}$ will decrease as training proceeds. This effect can be observed from the margin distribution graph in Figure 1 from Schapire *et al.* [86].¹ The “5,” “100,” and “1000” are the number of rounds. From round 100 to round 1000, the minimum margin (marked by the blue crosses) increases while the average margin decreases (the shaded red region).

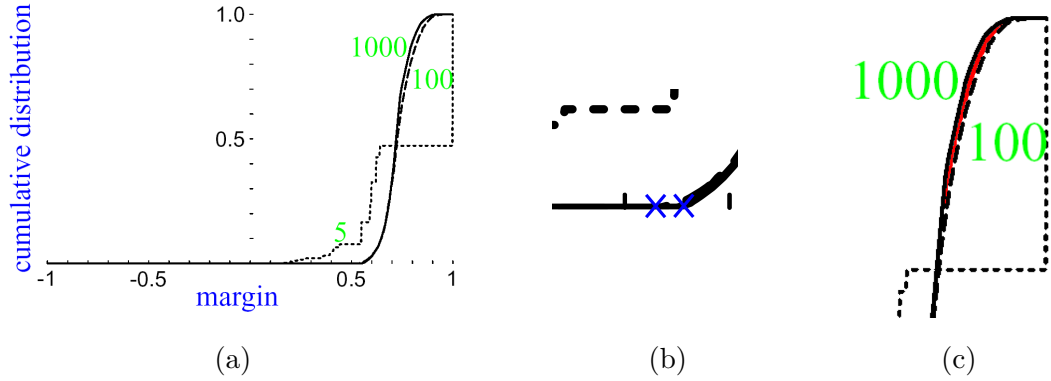


Figure 1: (a) The margin distribution graph in Schapire *et al.* [86]; (b) a zoom-in view of the increase in the minimum margin from round 100 to round 1,000, marked by blue crosses; and (c) a zoom-in view of the decrease in the average margin from round 100 to round 1,000, shaded by red.

3.2.2 Extension to Multiple Classes

Extension from the binary class case to the multiclass case is straightforward. One multiclass boosting algorithm is AdaBoost M2 [29], which constructs a set of ensembles, each representing one class, and the training maximizes the multiclass margin $M = H^{(y_i)} - \max \{H^{(y \neq y_i)}\}$ [2], where y_i is the correct class label. The minimum margin of AdaBoost M2 increases while the average margin decreases in the training, similar to the margins of binary class AdaBoost in training.

¹Permission to reprint figure granted by the Institute of Mathematical Statistics.

3.2.3 The Discriminative New Feature Space

SBHMMs *construct* a new feature space \mathcal{V} using the ensembles of boosting. We define \mathcal{V} as the output space spanned by the S ensembles $\mathcal{V} = (H^{(1)}, \dots, H^{(S)})$, in which S is the total number of the classes (states). For example, boosting constructs three ensembles, $H^{(1)}$, $H^{(2)}$, and $H^{(3)}$, corresponding to a three-class problem. Assuming that the outputs of the ensembles are $H^{(1)}(x) = 0.2$, $H^{(2)}(x) = 0.9$, and $H^{(3)}(x) = 0.7$, we will project x to the coordinate $(0.2, 0.9, 0.7)$ in \mathcal{V} . Note that (1) this projection is *nonlinear* because the ensemble is a combination of the nonlinear thresholding functions on the feature values and (2) the standard dimensionality reduction methods such as PCA can be applied to \mathcal{V} to further improve efficiency.²

The intuition of using \mathcal{V} as the new feature space is twofold: (1) The weights in the ensembles measure the importance of the (feature, decision stump) pairs, not just the features. This information is encoded in \mathcal{V} ; in contrast, ignoring the stumps and simply constructing a space by features with heavy weights is only a rough approximation of the feature selection results. (2) If assuming a Gaussian observation model, we explain that the margin properties refine the data distribution in \mathcal{V} to fit the HMMs as follows. The increase in the minimum margin and the decrease in the average margin indicate a natural clustering of data according to their labels in the output space of boosting [108]. We describe the intuition in a binary classification case. The output of the ensemble for binary classification can be mapped onto a one-dimensional axis. When this ensemble contains only one weak learner, the “+”s and “-”s are likely to be distributed everywhere from -1 to $+1$. As more and more weak learners are added, most of the “+” move to the positive side while most of the “-” move to the negative side of the axis. That is, the samples are *aggregated* in \mathcal{V} according to their labels.

Table 2 provides a simple example: When the conditional distribution of the feature value $p(\text{feature}|\text{class})$ is far from the Gaussian distribution, boosting can make the conditional distribution of the classification confidence $p(\text{output}|\text{class})$ more “Gaussian” in the

²In this dissertation, the results reported are without PCA.

output space of the ensembles. Note this change in the data distribution is desirable when we employ the Gaussian observation model, which is the most common choice in HMMs for recognition. We measure the skewness, the (unbiased) kurtosis, the z-test, and the Kolmogorov-Smirnov test to compare the Gaussianness of the data in the input space and the output space of boosting. For all four tests, a true Gaussian distribution should have a score of 0; and the lower the score, the closer the distribution to a true Gaussian. These four types of measurement are defined in Appendix B for reader’s convenience.

Further validations using the Georgia Tech Speech Reading (GTSR) data (details in Section 6.4) show that the original data are hyper-leptokurtic (high kurtosis), as those are in Table 2, so they may require a dense mixture of Gaussians. After the nonlinear projection by SBHMMs, the hyper-leptokurtic distribution also becomes more Gaussian in the new feature space, and the data kurtosis is reduced by 70% in our validation test (the details of which are presented in Section 6.4) due to the aggregation effect. The training of the (mixture of) Gaussian observation models is in turn improved.

Table 2: A simple example where AdaBoost can make linearly inseparable samples cluster according to their label in the output space while reducing the skewness, kurtosis and z-test values at the same time

Sample#	Feature	Class	Ensemble output (new feature)
1	1.0	+	+0.4551
2	2.0	+	+0.4551
3	3.0	+	+0.4551
4	4.0	+	+0.4551
5	5.0	+	+0.4551
6	6.0	-	-0.2625
7	7.0	-	-0.2625
8	8.0	-	-0.2625
9	9.0	-	-0.2625
10	10.0	+	+0.0547
11	15.0	-	-0.1454
12	20.0	-	-0.1454
13	30.0	+	+0.0823
14	40.0	-	-0.4551

Class	+		-	
Feature	Original	Output Space	Original	Output Space
Skewness	1.87	-0.95	0.97	-0.82
Kurtosis	1.75	-1.08	-0.38	-0.15
Z-test	0.71	-0.36	0.37	-0.31
KS-test	0.84	0.52	1.00	0.56

CHAPTER IV

DISCRIMINATIVE STATE-SPACE CLUSTERING

In the linguistic study of spoken languages, the words are decomposed to sub-word units such as syllables and phonemes. Such decompositions are essential to achieve scalability and accuracy in large vocabulary speech recognition [80]. Similarly, the search for linguistic sub-sign units for ASL has introduced the concept of ASL phonemes. We discuss ASL phonemes in Section 4.1. However, this chapter does not further investigate ASL linguistics nor the exact set of ASL phonemes. Instead, we focus on a data-driven decomposition of signs to improve feature selection and recognition accuracy in the rest of the chapter.

First, we briefly review the assumption made by the feature selection procedure proposed in the last chapter. The feature selection algorithm assumes that the basic learning units in sequence classification can be approximated by the states of the HMMs. The experimental validation in Chapter 6 indicates that this granularity is fine enough to represent the contrast of minimal pairs. This rough approximation is revisited in this chapter, and a data-driven sub-sign unit extraction algorithm, discriminative state-space clustering (DISC), is proposed to work jointly with SBHMMs in order to provide a more parsimonious representation. DISC automatically detects possible sharing of the learning units (states in SBHMMs) among different signs, and such units are then merged as basic building blocks (sub-sign units) for ASL signs to improve efficiency in recognition. Furthermore, the detection of the common building blocks avoids SBHMMs searching for the decision boundaries that are unlikely to exist, and thus reduces the risk of overfitting in discriminative feature selection (see Table 26).

4.1 ASL Phonemes

One major difference between ASL and English is that ASL has both sequential contrast and simultaneous contrast, while English only has sequential contrast [97]. The “cheremes” (phonemes) [94] proposed by Stokoe include handshape, location, and movement. The

Table 3: An example of a sequential contrast.

	GOOD		
	first part	middle part	last part
movement	hold	move out	hold
location	chin	transitional	out from chin
orientation	palm to chin	transitional	palm to chin
hand configuration	B	transitional	B
	BAD		
	first part	middle part	last part
movement	hold	move out	hold
location	chin	transitional	out from chin
orientation	palm to chin	transitional	palm to floor
hand configuration	B	transitional	B

Stokoe system primarily focuses on simultaneous contrast. An example is that the sign FATHER and MOTHER contrast in the chereme of location: the dominant hand (preferred hand) [97] touches the forehead and the chin in FATHER and MOTHER respectively, while the handshape and movement are the same for the two signs. However, the Stokoe system is ineffective at expressing the sequential details [56]. Its inadequacy for a sequential contrast, such as the difference between the minimal pairs GOOD and BAD in Table 3, was addressed by the movement-hold model [56], in which “handshape, rather than being a phoneme, is regarded as a set of (linguistic) features that partially define a segment (phoneme)” [55]. Those linguistic features for ASL are analogous to the linguistic features [51] in spoken languages. In speech, the concept of a linguistic feature is defined at the level of a simultaneous segmental contrast such as the comparison of /b/ and /p/ in Table 4; and the concept of a phoneme is defined at the level of a sequential contrast, such as /b/ and /p/ in BAT and PAT.¹ In such a definition, the location of contact (the forehead or the chin) is a simultaneous linguistic feature that separates the beginning of the compound signs BROTHER and SISTER. The latter part of the two signs, SAME, can be reused in modeling ASL.

This dissertation proposes to extract “machine-friendly” sub-sign units using the weakly-labeled (*i.e.*, no sub-sign labels available) training data based on the movement-hold model. This data-driven approach automatically constructs intermediate categories (sub-sign units)

¹See Appendix A for details.

Table 4: Articulatory linguistic features

	/p/	/b/
Point of articulation	bilabial	bilabial
Manner of articulation	stop	stop
Voiced or voiceless	voiceless	voiced

in order to directly reduce machine recognition errors, as opposed to extracting Signemes [70], the “core” part of signs for sign spotting. Using ASL decomposition to improve machine recognition has been proposed by Vogler and Metaxas [98]. The difference between their work and ours is that they define the entire set of phonemes manually according to the movement-hold model; in this work, we adopt a structure similar to that of the movement-hold model but construct the building blocks automatically in a data-driven manner to improve discrimination [112]. Traditional automatic phoneme extraction for ASL [7, 33] reduces the encoding complexity without explicit optimization for discriminative features, as opposed to the DISC-SBHMMs proposed in this dissertation.

4.2 Inseparable States as the Sub-sign Units

We review the intuition for the design of SBHMMs:

- The signs are sequentially composed by states.
- An individual set of features for each state is learned via feature selection.
- Discriminative learning ensures that the features are selected to maximize the separability of the states.
- The discrimination of the signs can be improved by the discrimination of the states that compose them.

In the previous chapter, we assume that different states of HMMs are building blocks of ASL. Therefore, the boosting algorithm searches for the features that separate every state of an HMM from all the other states of the same HMM and all the states of the other HMMs. In practice, this assumption may be violated in many cases. For example, the signs SON and DAUGHTER are composed of MALE-BABY and FEMALE-BABY respectively. The

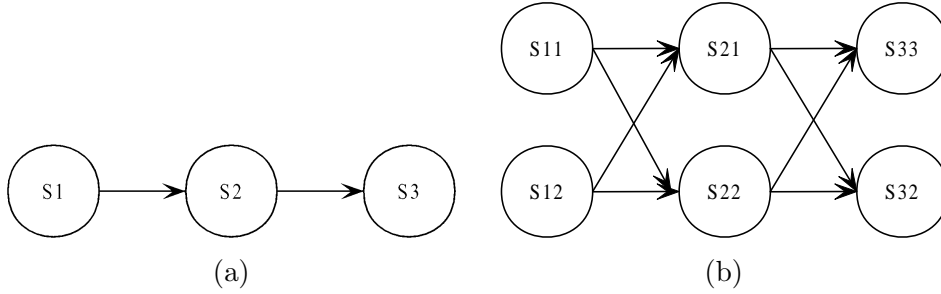


Figure 2: (a) an HMM composed of three states; (b) the same HMM after splitting now has 2^3 possible state transition paths.

two states that correspond to the later part of the sequences (BABY) of the two HMMs are likely to be similar and inseparable. Forcing boosting to select features that “separate” these two states will only lead to overfitting. Therefore, these states should be considered as one instead of two distinct building blocks of ASL. This dissertation uses the confusion matrix of the state classification to merge similar states for sub-sign unit extraction.

4.3 The State-Tying Paradigm

The traditional state-tying technique [80], which is successful at improving the efficiency and scalability of speech recognition, can be applied to this task. State tying has been typically applied to context-dependent states, but a recent study [22] in speech processing show that global tying across all states further improves accuracy. The state-tying procedure can be executed in two ways: top-down and bottom-up. The top-down approach, such as decision-tree state-tying, starts with a rough-scale representation and splits the states whose data distribution is not coherent. The samples whose likelihood falls below a specified threshold (in the “tail” of the distribution) are split from the state and used to estimate a new state. Such steps split certain states into two or more states. Then, the entire state space is retrained. This procedure is repeated until *none* of the leaf nodes can split or a predefined maximum number of states have been reached. Obviously, one major problem of the top-down approach is how to revise the state transition matrix to include the new states. As shown in Figure 2, a splitting of n states results in an exponential increase of 2^n possible state transition paths.

The second way for state-tying is the bottom-up approach. It starts with a fine-scale

representation and sequentially merges the states that are similar. The merging criterion is to minimize the reduction of data likelihood with a new, more compact model. Such merging is executed until the reduction of data likelihood reaches a predefined threshold or the number of states reaches a limit. The bottom-up state-tying has been widely used in practical speech recognition systems due to its simplicity [80]. We adopt the bottom-up approach in this dissertation.

4.4 The Iterative Clustering Algorithm

The discriminative state-space clustering (DISC) is executed as follows:

Algorithm 2 The DISC Algorithm

- 1: Randomly initialize the HMMs
 - 2: Run SBHMMs to select state-dependent discriminative features.
 - 3: **while** NOT (a pre-set lower bound P of the sub-sign units are extracted OR no state pair exceeds θ OR a pre-set number of rounds I are reached) **do**
 - 4: Extract the confusion matrix in separating the states.
 - 5: Use the Houtgast algorithm [4] to compute the similarity of each state pairs from the confusion matrix.
 - 6: Merge the top m most similar state pairs, or the state pairs whose similarity is above a threshold θ .
 - 7: Run SBHMMs to select state-dependent discriminative features in the new state space.
 - 8: **end while**
 - 9: Train and test in the new feature space computed by SBHMMs.
-

The experimental results in Chapter 6 suggest that DISC may decompose ASL into semantically meaningful pieces reducing the complexity of the model complexity without sacrificing its accuracy. Meanwhile, the DISC algorithm is subject to several limitations:

- The iterative optimization contains SBHMMs in the loop, which is costly in training.
- The horizon problem of greedy merging leads to a sub-optimal solution. The initialization point affects the final results (as all other EM algorithms do).
- The transitional movements (epenthesis [98]) may cause inaccurate clustering.
- If many new sign words are added to the training data, re-clustering is necessary.

- This optimization strategy cannot merge common state sequences.²

One alternative to the DISC algorithm is to follow the standard iterative state-tying algorithm in speech recognition for sub-sign unit extraction [80]. The standard benefit of using state-tying in speech recognition is to better employ limited training data for creating a complicated model. Here, state-tying also helps to select discriminative features more accurately in DISC. The major difference between the two is that DISC has a discriminative feature selection algorithm embedded in the training. We believe that this step is essential to the success of the tasks such as ASLR, in which a “good” set of features is not apparent to the user. For example, in our pilot study an unsupervised clustering of the speech reading data described in Chapter 6 showed meaningless clusters from the data cloud.

²The optimization strategy cannot recover from certain temporal over-segmentation. For example, DISC is capable of discovering that one state is necessary to model sequence AAAA, but it is unable to discover that, for two sequences ABCD and ABCE, three states A, B, and C can be (arguably) treated as one state. However, such over-segmentation should not cause serious problems in recognition.

CHAPTER V

DISCRIMINATIVE ANALYSIS FOR SIGN VERIFICATION

This chapter addresses the two challenges in a sign verification game called CopyCat. The first challenge is that sign verification must be optimized for both true positive and false positive rates, unlike traditional sign recognition, which is measured only by accuracy. The second challenge is that the sign verification game requires an efficient error analysis algorithm that facilitates the re-design of the game.

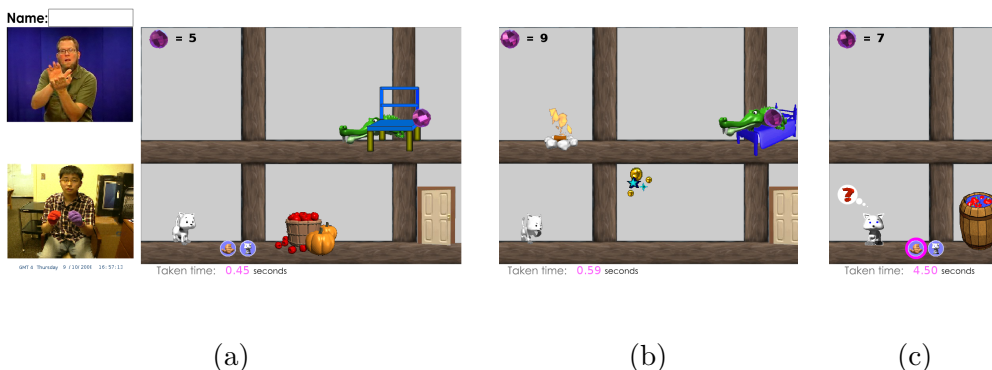


Figure 3: The CopyCat game. (a) When the “help” button is clicked, a tutor (top left) will show the user what to sign. (b) If the user signed correctly, the hero “poofs” the guard animal. (c) If the user signed incorrectly, the hero looks confused and encourages the player to sign again.

5.1 The CopyCat Project

5.1.1 CopyCat, the Game

The CopyCat project [54] at the Georgia Institute of Technology facilitates language skill acquisition for young deaf children through an interactive ASL-based game. During play, the children use sign language to interact with the hero in a “talking” scene in which they tell the hero where a “bad” animal guard is hiding. The guard animals are snakes, alligators, spiders, and “bad” cats. The guards are always hiding *in*, *on*, *under*, or *behind* various colored objects with a desired item close by. If the children know what to tell

Table 5: The vocabulary currently used in CopyCat

Category	Signs	Total
Subjects	ALLIGATOR, CAT, SNAKE, and SPIDER	4
Objects	BED, BOX, CHAIR, FLOWERS, WAGON, and WALL	6
Adjectives	BLACK, BLUE, GREEN, ORANGE, and WHITE	5
Prepositions	BEHIND, IN, ON, and UNDER	4

the hero regarding the location of the guards, such as “(a) BLUE ALLIGATOR ON (the) GREEN WALL” or “(a) BLACK SPIDER IN (the) WHITE FLOWERS,” they can push a “talk” button to turn the hero towards them so that they can sign to the hero. When they have finished signing, they push the “talk” button again to provide timing cues for the sign language recognition data being collected via video camera. If the children are uncertain of what to sign, they can click a “help” button to see a tutor (in the top left corner of the screen) telling them what to say, as shown in Figure 3. The child may view the tutor repeatedly if he/she so chooses. After the children sign, the automatic verifier, a specialized ASL recognizer, compares the input against the groundtruth phrase. If the signing is correct, the hero will “poof” the guard animal and retrieve the hidden item; otherwise, the player will be prompted to sign again. The signs and phrases currently used in CopyCat are summarized in Tables 5, 6, 7, and 8. The vocabulary size and the number of phrases are very small for a ASL learning task; however, note that (1) the purpose of CopyCat is to help children develop short-term memory and obtain some basic language skills and (2) methods discussed in this chapter are scalable to more complicated games.

5.1.2 Data Collection

This section explains the data collection details used for sign verification and phrase selection for the CopyCat game. We collect the signing of the game player by one video camera and a pair of colored-gloves with accelerometers attached. In order to achieve a relatively stable video input for recognition, we fixed the position of the camera and the chair of the game player on the floor, as shown in Figure 4.¹ A game player controls the data capture by clicking the “talk” button before and after his/her signing. The signing is verified as

¹Figures 4, 5, 6, and 7 are courtesy of Zahoor *et al.* [113].

Table 6: The three-sign phrases currently used in CopyCat

Index	Phrase
1	ALLIGATOR BEHIND WALL
2	SNAKE BEHIND WALL
3	SPIDER ON WALL
4	CAT ON WALL
5	SNAKE UNDER CHAIR
6	SPIDER ON CHAIR
7	ALLIGATOR BEHIND CHAIR
8	CAT UNDER CHAIR
9	ALLIGATOR IN BOX
10	SPIDER IN BOX
11	CAT BEHIND BOX
12	SNAKE ON BOX
13	CAT BEHIND BED
14	ALLIGATOR ON BED
15	SNAKE UNDER BED
16	SPIDER UNDER BED
17	ALLIGATOR IN WAGON
18	SPIDER UNDER WAGON
19	CAT BEHIND FLOWERS
20	SNAKE IN FLOWERS

Table 7: The four-sign phrases currently used in CopyCat

Index	Phrase
21	ALLIGATOR ON BLUE WALL
22	SPIDER IN GREEN BOX
23	SPIDER IN ORANGE FLOWERS
24	SNAKE UNDER BLUE CHAIR
25	ALLIGATOR BEHIND BLUE WAGON
26	SNAKE UNDER BLACK CHAIR
27	CAT ON BLUE BED
28	CAT ON GREEN WALL
29	ALLIGATOR UNDER GREEN BED
30	SPIDER ON WHITE WALL
31	SPIDER UNDER BLUE CHAIR
32	ALLIGATOR IN ORANGE FLOWERS
33	CAT BEHIND ORANGE BED
34	ALLIGATOR BEHIND BLACK WALL
35	SNAKE UNDER BLUE FLOWERS
36	CAT UNDER ORANGE CHAIR
37	SNAKE IN GREEN WAGON
38	SPIDER IN BLUE BOX
39	ALLIGATOR BEHIND ORANGE WAGON
40	CAT UNDER BLUE BED

Table 8: The five-sign phrases currently used in CopyCat.

Index	Phrase
41	BLUE ALLIGATOR ON GREEN WALL
42	ORANGE SPIDER IN GREEN BOX
43	BLACK SNAKE UNDER BLUE CHAIR
44	BLACK ALLIGATOR BEHIND ORANGE WAGON
45	GREEN SNAKE UNDER BLUE CHAIR
46	BLACK SPIDER IN WHITE FLOWERS
47	BLACK CAT ON GREEN BED
48	WHITE CAT ON ORANGE WALL
49	GREEN ALLIGATOR UNDER BLUE FLOWERS
50	BLUE SPIDER ON GREEN BOX
51	ORANGE ALLIGATOR IN GREEN FLOWERS
52	BLACK CAT BEHIND GREEN BED
53	WHITE ALLIGATOR ON BLUE WALL
54	ORANGE SNAKE UNDER BLUE FLOWERS
55	GREEN SPIDER UNDER ORANGE CHAIR
56	BLACK CAT IN BLUE WAGON
57	WHITE CAT IN GREEN BOX
58	WHITE SNAKE IN BLUE FLOWERS
59	ORANGE SPIDER UNDER GREEN FLOWERS

“good” (match) or “bad” (no-match) by a verifier (manually by human or automatically by computer). In order to collect training and testing data for our algorithm, we invited an ASL linguist as a human wizard to provide this label in a “wizard of Oz” setting [54]. He labeled 1,467 phrases in the CopyCat dataset as “good.” However, a human wizard may make mistakes. We then manually inspect the results of this “human verifier.” The consensus of the two steps finally determined that 1,077 phrases “match” the groundtruth script with good quality for training and testing, and the rest, either of in agreement on “bad” or in disagreement in the two aforementioned steps, as “no-match” for testing (because we do not need “no-match” for training). Table 9 lists the metric (performance) achieved by the human verifier.

The game begins with a level of three-sign phrases. Once the player signs four correct phrases in succession, the game increases difficulty from three- to four-sign phrases or from four- to five-sign phrases. The game ends when the player has finished a total of 20 correct phrases. This setting ensures that we have a sufficient amount of good data for training and testing.

Table 9: Accuracy metric achieved by the human verifier in CopyCat (courtesy of Zahoor Zafrulla and Harley Hamilton)

Human Verifier	%
True Positive	98.57
True Negative	68.67
False Positive	31.33
False Negative	1.43
Overall Accuracy	90.43

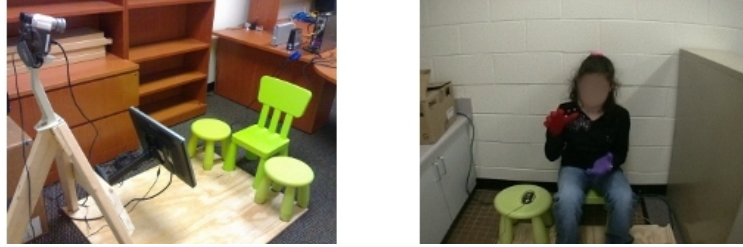


Figure 4: Left: Game kiosk setup. Note that the camera and the chair of the player are fixed on the floor. Right: a sample video frame captured by the camera.

5.1.3 Features

Figure 5 shows the colored data glove with accelerometers attached. We compute 28 visual features from the 720x480 video frame at 20 Hz and six acceleration features from two three-axis accelerometers. We first convert the video frame to HSV space and then track the hands using hue (H) information. The ten visual features for each hand are: the $\Delta x, \Delta y$ of the centroid, the size of the blob, the length of the major and minor axes, the eccentricity, the orientation of the major axis, and the change in the orientation of the major axis [11]. We use a cascade of boosting classifiers to track the head and compute the features with regard to the relative positions of the head and the hands, shown in Figure 6, as $\alpha_1, \alpha_2, \alpha_3, l_1, l_2, l_3, \theta_l$, and θ_r . For the sake of simplicity, we did not include facial expression. We also recorded six accelerometer features, that is, two accelerometer readings (one on each glove), from -2g to +2g, of the 3D acceleration. Preliminary experiments suggest that accelerometer features do not yield a significant boost in the accuracy of recognition/verification accuracy [113]. In the experimental results described below, we use only the 28 visual features.



Figure 5: Left: a pair of colored data gloves. Right: each glove has one 3D accelerometer.

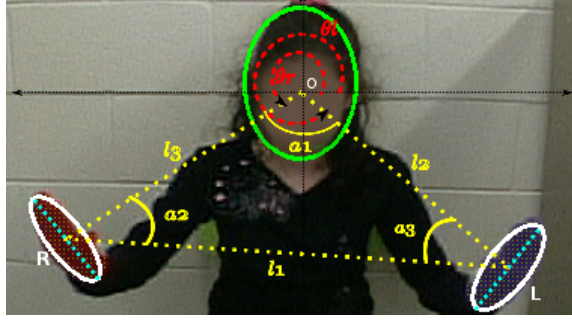


Figure 6: Visual features for CopyCat

5.1.4 The CopyCat Dataset

We collected signing data from 18 children. The data collection is separated into three parts. The first part contained the signing of five children (child#1 to child#5), collected in the fall of 2008 and denoted as Fall08; the second part contained the signing of seven children (child#6 to child#12), collected in the early spring of 2009 and denoted as Spring09I; the third part contained the signing of 16 children (child#3 to child#18), collected in late spring of 2009 and denoted as Spring09II. The children who participated in these studies exhibited three kinds of hand dominance: left-hand dominance, right-hand dominance, and mixed dominance. Of the aforementioned “good” phrases, 698 are right-hand dominant, 151 are left-hand dominant, and 228 are mixed. The children also made several variations of every sign, listed in Table 10. Some variations of the signing are shown in Figure 7. The variations, difficult to model by a single HMM per sign, are constructed as separate sign tokens in recognition. Thus, the CopyCat dataset contains 45 tokens. To include all signing variations (tokens) in training, we decided to use Spring09II plus nine phrases from Spring09I, denoted as Spring09II+ for training, Fall08 for parameter tuning, and Spring09I minus the nine phrases from training, denoted as Spring09I-, for testing. This



Figure 7: Some signing variations. Left: BED_RIGHT and BED_LEFT. Right: CAT_TWO_HANDED and CAT_RIGHT.

training/testing split is fixed for all the related experiments presented in this dissertation. While the training and testing data are independent, some children participate in both. Some children are unique to the testing set.

5.1.5 The CopyCat-Adult Dataset

Due to the high signing variance in the CopyCat dataset, we recorded the signing of eight adult Georgia Tech participants who were non-native signers. The signing variance was significantly lower in this dataset than in CopyCat dataset, so we keep only one common signing pattern (token) per sign and discard the recordings with rare variations. Therefore, CopyCat-Adult contains 492 phrases and only 19 tokens. For the CopyCat-Adult dataset, we randomly picked 80% of the data for training and 20% of the data for testing. The training/testing split is fixed for all the related experiments presented in this dissertation.

5.2 Sign Verification

Automatic sign verification is integral to the CopyCat game. Speech/utterance verification is a process of verifying “the claimed content of a spoken utterance” [52]. Similarly, this dissertation defines “sign verification”² as a process of verifying the claimed content of the signs. Let us define an ideal verifier (\mathcal{V} stands for verifier)

$$\mathcal{V}_{\text{ideal}}(P, F_Q) = \begin{cases} 1 & \iff P = Q \\ 0 & \iff P \neq Q, \end{cases} \quad (4)$$

²Sign verification in this dissertation compares input with a groundtruth script, which is different than the verification step to prune spotting results [3, 106]. Sign verification should also be distinguished from signer verification, which is analogous to speaker verification to verify identity.

Table 10: Variations of the signs in the CopyCat dataset

Token index	Sign	Variation
1	ALLIGATOR_TWO_HANDED	ALLIGATOR
2	ALLIGATOR_LEFT	
3	ALLIGATOR_RIGHT	
4	CAT_TWO_HANDED	CAT
5	CAT_LEFT	
6	CAT_RIGHT	
7	SNAKE_LEFT	SNAKE
8	SNAKE_RIGHT	
9	SPIDER_LEFT	SPIDER
10	SPIDER_RIGHT	
11	BED_LEFT	BED
12	BED_RIGHT	
13	BOX	BOX
14	BOX_CLEAR	
15	CHAIR_LEFT	CHAIR
16	CHAIR_RIGHT	
17	FLOWER_LEFT_LEFT	FLOWER
18	FLOWER_LEFT_RIGHT	
19	FLOWER_RIGHT_LEFT	
20	FLOWER_RIGHT_RIGHT	
21	WAGON_RIGHT_UP	WAGON
22	WAGON_RIGHT_DOWN	
23	WAGON_LEFT_UP	
24	WAGON_LEFT_DOWN	
25	WAGON_RIGHT_TWO	
26	WALL_BACK	WALL
27	WALL_FRONT	
28	BEHIND_LEFT	BEHIND
29	BEHIND_RIGHT	
30	IN_LEFT	IN
31	IN_RIGHT	
32	ON_LEFT	ON
33	ON_RIGHT	
34	UNDER_LEFT	UNDER
35	UNDER_RIGHT	
36	BLACK_LEFT	BLACK
37	BLACK_RIGHT	
38	BLUE_LEFT	BLUE
39	BLUE_RIGHT	
40	GREEN_LEFT	GREEN
41	GREEN_RIGHT	
42	ORANGE_LEFT	ORANGE
43	ORANGE_RIGHT	
44	WHITE_LEFT	WHITE
45	WHITE_RIGHT	

Table 11: True positive, false positive, true negative, false negative, and accuracy rates are not independent for ASL verification.

	Groundtruth Positive	Groundtruth Negative
Predicted Positive	True Positives	False Positives
Predicted Negative	False Negatives	True Negatives

True Positive Rate	$\frac{\text{number of True Positives}}{\text{number of groundtruth positive instances}}$
False Positive Rate	$\frac{\text{number of False Positives}}{\text{number of groundtruth negative instances}}$
True Negative Rate	$\frac{\text{number of True Negatives}}{\text{number of groundtruth negative instances}} = 1 - \text{False Positive Rate}$
False Negative Rate	$\frac{\text{number of False Negatives}}{\text{number of groundtruth positive instances}} = 1 - \text{True Positive Rate}$
Accuracy	$\frac{\text{number of True Positives} + \text{number of True Negatives}}{\text{number of instances}}$

in which P is an input groundtruth phrase (claimed content), such as ALLIGATOR ON WALL, and F_Q is a signing sequence $F = \langle f_1, f_2, \dots, f_{T-1}, f_T \rangle$ corresponding to phrase Q , which can have the same or a different number of signs than P , or could even be some random gestures (noise). We implement $\mathcal{V}(P, F_Q)$ using the likelihood of $L(P, F_Q) = \text{Prob}(F_Q|P)$:

$$\mathcal{V}_{\text{ideal}}(P, F_Q) = \begin{cases} 1 & \iff L(P, F_Q) \geq \theta \\ 0 & \iff L(P, F_Q) < \theta, \end{cases} \quad (5)$$

in which theta is a similarity threshold usually set empirically by additional parameter validation.

Compared with traditional ASL recognition, $\mathcal{R}(F_Q)$ (\mathcal{R} stands for recognizer) is defined as

$$\mathcal{R}(F_Q) = P^* = \arg \max_{P \in \{P_i\}} \{L(P, F_Q)\} \quad (6)$$

ASL verification $\mathcal{V}(P, F_Q)$ takes not only the input of the signing F_Q but also the “groundtruth phrases” P . ASL recognition classifies a valid signing sequence into one of the known sign labels P_i , and this process can be evaluated by a single metric of recognition accuracy. ASL verification in this dissertation, on the other hand, determines whether an (maybe invalid) input signing F_Q and an input phrase label P match. Such a task can be evaluated by five metrics: a true positive rate, a false positive rate, a true negative rate, a false negative rate, and accuracy, but they are not independent, as shown in Table 11.

One straightforward algorithm is to use recognition for verification: that is, first compute

ASLR $P^* = \mathcal{R}(F_Q)$ for the input signing sequences F_Q , and then compare the recognition results P^* with the groundtruth labeling P . However, this algorithm neglects the groundtruth labeling P as an important source of prior knowledge. When the game scene shows $P = \text{BLUE ALLIGATOR ON GREEN WALL}$, a child may make a mistake on one sign or two, but the signing is very unlikely to be, for example, $\text{ORANGE SPIDER UNDER WHITE BOX}$ if we assume that the child plays the game responsibly. In ASL verification, while additional information P is available, we can measure the similarity between the current input F_Q and other empirical signing data corresponding to P . However, a naïve algorithm that computes only similarity will become a “forced alignment” [80] and produce trivial results since even random noise can be aligned with any phrase script. Therefore, the first challenge is to find an optimal threshold θ in Equation 5 for the similarity measurement of match versus no-match. Such a threshold θ can be determined empirically by a brute force search on a separate validation set. This dissertation, on the other hand, proposes automatically computing one threshold for each sign.

The idea to automatically “select” thresholds comes from the literature of feature selection. In automatic feature selection, randomly-generated features that are irrelevant to the class label of the samples are added to the pool of features. These random features are called “probes,” because once the feature selection algorithm picks any of these probes as the most informative feature, the selection stops. Inspired by the idea of probe features, this dissertation proposes the idea of “probe inputs.” Randomly-generated input sequences are supplied to the classification model trained by correct signings, and the similarity (likelihood) of the random data to the trained model serves as the threshold for verification. Intuitively, signings are rejected as “no-match” if their likelihoods are no higher than those of the random inputs. The experiments in Chapter 6 show that the thresholds computed by such an intuitive strategy provide comparable results with empirically determined thresholds using exhaustive search.

The merit of using probe instead of exhaustive search is fast online adaptation. Although brute force search on a ROC curve computed from some training data can be automated by a script; (1) this search is costly and thus must be executed offline, (2) this search

assumes that the offline training data adequately represent the testing data, and (3) a user-independent verifier has to make a trade-off when selecting a fixed threshold. For example, if the signing from one child has a higher likelihood for matching than that from another child, a single, fixed threshold will either create unnecessary false positives for the first child or false negatives for the second child. In comparison, the probe technique can be easily implemented as an online process and can be applied to multiple users.

5.2.1 Recognizer versus Verifier

A verifier generalizes traditional recognizers $\mathcal{R}(F_Q) = \arg \max_P \{L(P, F_Q)\}$. An ideal recognizer observes $Q = P^* = \arg \max_P \{L(P, F_Q)\}$, that is, the script that matches the input is the true script signed. Meanwhile, the probe-based verifier is actually $\mathcal{V}(P, F_Q) = \mathbf{1}(P = Q^* = \arg \max_Q \{L(P, F_Q)\})$, that is, the signing that best matches the script is the true input instead of other random signings.

5.3 Phrase Selection for Game Re-design

This section addresses the second challenge in CopyCat, phrase selection. In this scenario, the game designer has complete power to choose the verification phrases of the game. As described in Chapter 1, the choices of the phrases do not have to include the entire ASL vocabulary but are required to interact with the user (deaf children) at an acceptable accuracy. In speech recognition, if the accuracy is higher than 95%, most users are willing to repeat their utterance in case of error [80]; otherwise, the users become impatient and tend to abandon the automatic system. Similarly, a low verification accuracy may discourage users. For example, in one of our pilot studies, one deaf child, after being prompted with the confused cat several times, signed the phrase “stupid cat” to the camera.

Therefore, the success of CopyCat depends on a high true positive rate in verification. The true positive rate can be improved by selecting the phrases that are the “most distinguishable” to the verifier: among the correct signings by children, these phrases are most accurately identified as a “match” by the verifier $\mathcal{V}(P, F_P)$. Therefore, in phrase selection, the true positive rate is the same as the classification accuracy, and a verifier is the same as a recognizer. We use these terms interchangeably in the rest of the chapter, which describes

Table 12: The four phrases for a simplified game, mini-CopyCat, ranked by an artificial testing accuracy

Subject	Preposition	Object	Testing Error %	Rank
SPIDER	ON	WALL	5%	1
ALLIGATOR	UNDER	WALL	15%	2
ALLIGATOR	ON	BOX	30%	3
SPIDER	ON	BOX	35%	4

the selection procedure in detail.

The syntax used by CopyCat is restricted: Each phrase is composed by an optional adjective, a subject, a preposition, an optional adjective, and an object, in order, as shown in Tables 5, 6, 7, and 8. Given a set of signs, such as those in Table 5, the phrase selection algorithm should pick a total of k triplets, quadruplets, or quintuplets. To achieve high recognition accuracy, we should reduce two types of ambiguity. The first is ambiguity between signs in the same grammatical location of the phrase, *e.g.*, GREEN and BLUE, which usually causes substitution errors when both signs are in the vocabulary. Therefore, we should either introduce new features to disambiguate this confusing pair or remove at least one sign of the pair from the vocabulary to ensure recognition quality. In fact, mis-alignment may cause substitution errors as well. However, such errors are unlikely because (1) in a sign verification task, we expect the recognizer to have reasonably high accuracy already; and (2) the alignment problem is further averted when the second type of ambiguity, which results from co-articulation between two adjacent signs, diminishes. The second type of ambiguity can be reduced by carefully selecting the combination of the signs in the phrases used.

5.4 Phrase Selection

To explain clearly the minimization of co-articulation errors by using phrase selection, this section uses a simplified game, called mini-CopyCat, which contains only four three-sign phrases shown in Table 12. Note that in this section many values, such as the testing accuracy, are artificial for illustration purposes.

Table 13: All eight possible phrases for mini-CopyCat ranked by their artificial testing accuracy

Subject	Preposition	Object	Testing Error %	Rank
ALLIGATOR	ON	WALL	0%	1
SPIDER	ON	WALL	5%	2
SPIDER	UNDER	WALL	10%	3
ALLIGATOR	UNDER	WALL	15%	4
SPIDER	UNDER	BOX	20%	5
ALLIGATOR	UNDER	BOX	25%	6
ALLIGATOR	ON	BOX	30%	7
SPIDER	ON	BOX	35%	8

5.4.1 Selection by Classification Accuracy

If we assume that each phrase in Table 12 occurs 100 times, the average testing error for mini-CopyCat is $(100 * 5\% + 100 * 15\% + 100 * 30\% + 100 * 35\%)/400 = 22\%$. A simple way to improve accuracy is to remove the least accurate phrase, SPIDER ON BOX, from the game. The new average error is $(100 * 5\% + 100 * 15\% + 100 * 30\%)/300 = 17\%$. However, this approach also reduces the complexity of the game complexity by 25%. Can we improve the classification accuracy with four phrases in the game? The answer is “yes, we can.” If we test the accuracy of more phrases that can be constructed using the six signs, we will be able to select the four phrases with best testing accuracy among them, and thus improve the average accuracy and create a better gaming experience. In an extreme case, if we test all eight combinations by subject = {ALLIGATOR, SPIDER}, preposition = {ON, UNDER}, and object = {WALL, BOX}, as shown in Table 13, we will be able to select the “best” four phrases for the game.

However, in order to obtain the “best” four phrases, we need to build a mini-CopyCat game with all eight phrases and have the children test all of them in the game. Such an exhaustive approach is intractable in real-world game design. In fact, the current version of CopyCat will yield $4*4*6+4*6*5*4+5*4*6*5*4=2976$ phrases to be tested. If we assume that there are n signs per category, and the length of the phrases is m , the combinations are on a scale of $O(n^m)$, which is exponential in m . The next section presents two algorithms that can reduce the complexity to quadratic $O(n^2 \cdot m)$.

Table 14: Artificial likelihoods for the preposition signs in the five sessions of mini-CopyCat

Subject	Preposition	Log-likelihood	Object
ALLIGATOR	ON	-0.2	BOX
SPIDER	ON	-0.5	BOX
ALLIGATOR	ON	-0.3	BOX
ALLIGATOR	ON	-0.4	BOX
SPIDER	ON	-0.7	BOX

5.4.2 Bi-gram Predictor for Phrase Accuracy

The goal of our prediction is to approximate the phrase ranking of the true testing error by the ranking of a score that is easy to compute. In order to predict the *relative ranking* of the testing accuracy of *unseen* phrases, this dissertation proposes a prediction technique based on partial information (segments) of seen phrases, called the bi-gram error-ranking (BIG) predictor. Since the likelihood for each sign in the phrase indicates how similar the signing is to the model, we can measure the co-articulation effect by computing how much the previous sign affects the likelihood of the one that follows. Let us assume that Table 14 contains the log-likelihoods of the preposition signs measured in five sessions of mini-CopyCat.

Then we are able to compute the average log-likelihoods for ON in a particular combination and all combinations:

$$\bar{L}L(\text{ON}|\text{ALLIGATOR}) = [(-0.2) + (-0.3) + (-0.4)] / 3 = -0.3.$$

$$\bar{L}L(\text{ON}|\text{SPIDER}) = [(-0.5) + (-0.7)] / 2 = -0.6.$$

$$\begin{aligned} \bar{L}L(\text{ON}) &= \bar{L}L(\text{ON}|\text{ALLIGATOR}) * P(\text{ALLIGATOR}) + \bar{L}L(\text{ON}|\text{SPIDER}) * P(\text{SPIDER}) = \\ &(-0.3) * 3/5 + (-0.6) * 2/5 = -0.42. \end{aligned}$$

Therefore, we define the “cost” to have ON after ALLIGATOR as the difference between the average log-likelihoods of ON for this particular combination and all combinations:

$$C(\text{ALLIGATOR}, \text{ON}) = \bar{L}L(\text{ON}) - \bar{L}L(\text{ON}|\text{ALLIGATOR}) = (-0.42) - (-0.3) = -0.12,$$

and the “cost” to have ON after spider as

$$C(\text{SPIDER}, \text{ON}) = \bar{L}L(\text{ON}) - \bar{L}L(\text{ON}|\text{SPIDER}) = (-0.42) - (-0.6) = 0.18.$$

If we define the phrase score for ALLIGATOR ON BOX as

$$S(\text{ALLIGATOR}, \text{ON}, \text{BOX}) = C(\text{ALLIGATOR}) + C(\text{ALLIGATOR}, \text{ON}) + C(\text{ON}, \text{BOX}),$$

then we can then rank all the phrases according to the phrase score in an ascending order. Note that the first sign has no preceding signs, so the cost is defined as its marginal log-likelihood.

5.4.3 Uni-gram Predictor for Phrase Accuracy

For comparison, this dissertation also computes a uni-gram predictor for phrase accuracy. A uni-gram predictor is similar to a bi-gram predictor except that the “cost” of a sign is solely determined by the unconditioned likelihood without context information. For example $S(\text{ALLIGATOR}) = \bar{L}L(\text{ALLIGATOR})$. A uni-gram predictor, being a simplified version of a bi-gram predictor, simulates the sequential likelihood computation by a uni-gram observation model of the underlying graphic model (a hidden Markov model in this dissertation) without a transition model. The computational complexity to learn a uni-gram model is linear $O(n)$. The advantages of using a uni-gram predictor are that it is less complex and less prone to noise or lack of data in the transition model, while the greatest disadvantages of using it is that the uni-gram model is unable to address co-articulation, if it exists.

5.4.4 Measurement of the Quality of Prediction

This section describes how to measure the quality of the approximation. In information retrieval, a standard way of evaluating *one* ranking function is normalized discounted cumulative gain (nDCG) [37]. Here, we compare *two* ranking functions. This dissertation compares the ranking results by Spearman footrule distance (Manhattan distance) [45], described as follows.

Assume that we have two ranking functions $f^a(\cdot)$ and $f^b(\cdot)$; the set of phrases is $\mathcal{Z} = \{z_1, z_2, \dots, z_{n_z}\}$. The rankings are $\mathcal{R}^a = \{r_i^a | f^a(s_i) = r_i^a\}$ and $\mathcal{R}^b = \{r_i^b | f^b(s_i) = r_i^b\}$. Define the “distance” between the two ranking results as the L_1 distance between two vectors \mathcal{R}^a and \mathcal{R}^b ,

$$DL_1(a, b) = \sum_{i=1}^{n_z} |r_i^a - r_i^b|. \quad (7)$$

The average distance is

$$aDL_1(a, b) = DL_1(a, b)/n_z. \quad (8)$$

Other alternatives to measure the difference of two ranking functions include Kendal tau distance [45] and Pearson product-moment correlation coefficient (PMCC) [101]. Kendal tau distance computes the pairwise disagreements between two rankings \mathcal{R}^a and \mathcal{R}^b , as shown in Equation 9. Let $n = ||\mathcal{R}^a|| = ||\mathcal{R}^b||$, then this measurement equals zero when \mathcal{R}^a and \mathcal{R}^b are identical and $n_z * (n_z - 1)/2$ when \mathcal{R}^a is in reverse order of \mathcal{R}^b . Kendal tau distance is usually normalized by $n_z * (n_z - 1)/2$ so that it ranges between zero and one. PMCC computes the linear correlation between two sets of data points \mathcal{R}^a and \mathcal{R}^b , as shown in Equation 10, in which μ_a and μ_b are the mean value of \mathcal{R}^a and \mathcal{R}^b . $\rho = 0$ when \mathcal{R}^a and \mathcal{R}^b have no correlation; and $\rho = \pm 1$ when \mathcal{R}^a and \mathcal{R}^b are in perfect positive/negative linear correlation.

$$K(a, b) = \sum_{\forall i, j} \mathbf{1}(r_i^a < r_j^a \text{ and } r_i^b \geq r_j^b). \quad (9)$$

$$\rho = \frac{\sum (r_i^a - \mu_a)(r_i^b - \mu_b)/(n - 1)}{\sqrt{\sum (r_i^a - \mu_a)^2 \sum (r_i^b - \mu_b)^2}}. \quad (10)$$

5.4.5 Selection for Unseen Phrases

We are able to compute the score of any unseen phrases as long as they are composed by seen bi-grams using BIG in polynomial time $O(n^2 \cdot m)$. However, because there are $O(n^m)$ phrases, computing the ranking of all unseen phrases still takes exponential time. This section claims that selecting the top k phrases from the ranking can be reduced to a “k-shortest path problem” in graph theory and solved in polynomial time by Yen’s algorithm [107].

5.4.5.1 The k -shortest Path Problem

Given a directed graph $G(V, E, W)$, V is a set of vertices (signs in CopyCat) and E is a set of directed edges (the adjacency ordering or grammar of signs) with W as their weight (cost computed in the previous section), a path \mathcal{P} (phrase) starting at $v(i_1)$ and ending at $v(i_l)$ is a sequence of vertices $\mathcal{P} = (v(i_1), v(i_2), \dots, v(i_l))$, connected by edges in E , and the distance of the path $w(\mathcal{P})$ (phrase score) is the sum of the weight of the connected edges. The shortest path problem finds $\mathcal{P}^* = \arg \min w(\mathcal{P})$ (lowest phrase score). This problem can be solved easily by Dijkstra’s algorithm [21]. The k -shortest path problem is to find k paths $\mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^k$ from all the paths that start at vertex $v(s)$ and end at vertex $v(t)$, such that $w(\mathcal{P}^1) \leq w(\mathcal{P}^2) \leq \dots \leq w(\mathcal{P}^k) \leq w(\mathcal{P}^\#)$, for $\forall \mathcal{P}^\# \neq \mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^k$. Numerous studies [26, 60, 107] have been published on this subject for polynomial time solutions, most of which are *deviation algorithms*.

Deviation algorithms search for the k -shortest paths based on the construction of a “pseudo”-tree. The root of the tree is the start node, and the leaves are some end nodes. Each sub-optimal path is a “deviation” from the optimal (shortest) path. For example, Figure 8(e) is the pseudo-tree for the k -shortest path ($k = 4$) from A to D in the graph shown in Figure 8(a). The tree is “pseudo” in the sense that it can contain repeated nodes, yet the multiple appearance of the same node can be distinguished by its association with different paths. The intuition of the deviation algorithm is that the $(k+1)^{th}$ shortest path can be created by deviation from the k^{th} shortest path with proper restrictions. For example, the pseudo-tree in Figure 8(e) can be constructed by the following steps: compute the shortest path from A to D by the Dijkstra algorithm [21], shown in Figure 8(b); then compute the second shortest path by choosing the shorter of the two deviation paths (one at node B and the other at node A) shown in Figure 8(c); similarly, we can compute the third-shortest and fourth shortest paths by deviation, shown in Figures 8(d) and (e). This dissertation employs a variation of Yen’s algorithm [60] to compute the k -shortest path in $O(kn(m + n \log n))$, in which $n = |V|$ and $m = |E|$. Although this variation shares the same bound as the original Yen’s algorithm [107], it performs more efficiently in practice.

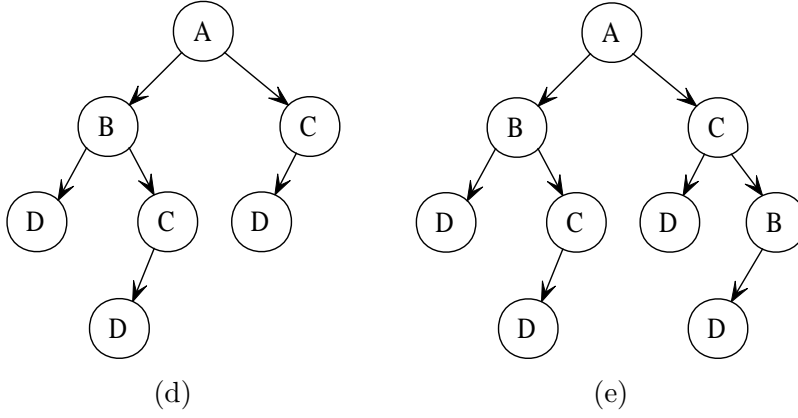
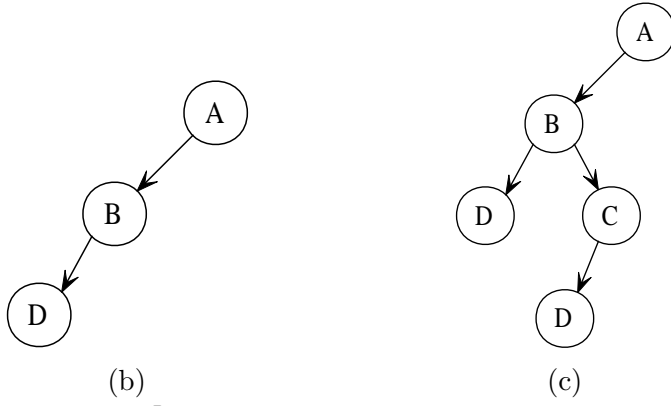
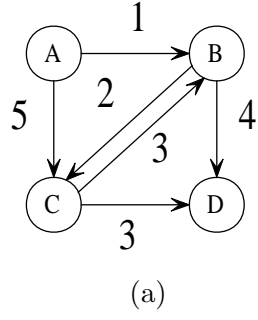


Figure 8: (a) a simple example of graph; (b),(c),(d), and (e), the construction of the pseudo-tree for (a).

CHAPTER VI

EXPERIMENTAL VALIDATIONS

To validate the claims about segmental features in Chapter 3, we first conduct a synthetic experiment, in which we show that feature selection in a segmental manner is the key to improving discrimination when the time sequences have different informative features at different phases. Then we evaluate the performance of the SBHMMs across the four domains and compare it with that reported previously by other researchers. Next we show some results of the performance of DISC-SBHMMs. Note that the size of the boosting ensembles is empirically determined for each application. Finally, we show the results of the probe technique and the bi-gram error ranking predictor (BIG) for sign verification in CopyCat and CopyCat-Adult dataset.

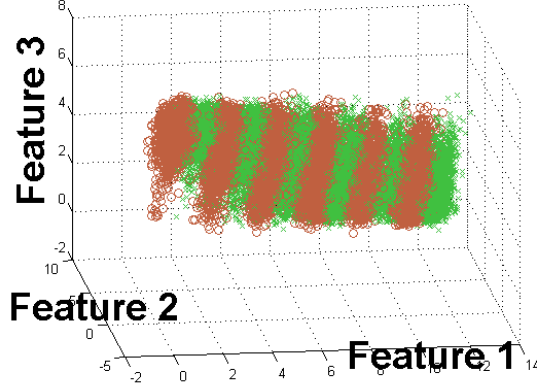
6.1 The Synthetic Experiment

We construct two six-state left-to-right HMMs (sequence generators), with the mean of the Gaussian observation model listed in Table 15. The covariance of the Gaussian is diagonal, and the variance of each dimension is fixed at 0.1. We sample 2,000 sequences of length 12 from each of the two HMMs. In the three-dimensional observation vectors, the informative feature is highlighted in blue in the table. The other two dimensions contain irrelevant dynamics. For training, we randomly pick 1,000 sequences generated by HMM₁ and 1,000 by HMM₂ and use the remaining 2,000 sequences for testing. Because the data are truly generated by HMMs, and feature 1 provides a clean indication of the class label, the two HMMs (sequence recognizers) trained on these data produce a perfect (0% error) recognition rate. We call this the “clean” experiment.

We then experiment using sequences with different informative features at different phases (states). For the 4,000 sequences obtained, we swap feature 1 with feature 3 in the first three states while maintaining the other parameters the same as before, shown in Table 16. Note that because the three features are still conditionally independent of each

Table 15: Synthetic example (clean)

model	feature	state 1	state 2	state 3	state 4	state 5	state 6
HMM ₁ (orange)	f1	1	3	5	7	9	11
	f2	1	2	3	4	5	6
	f3	6	5	4	3	2	1
HMM ₂ (green)	f1	2	4	6	8	10	12
	f2	1	2	3	4	5	6
	f3	6	5	4	3	2	1

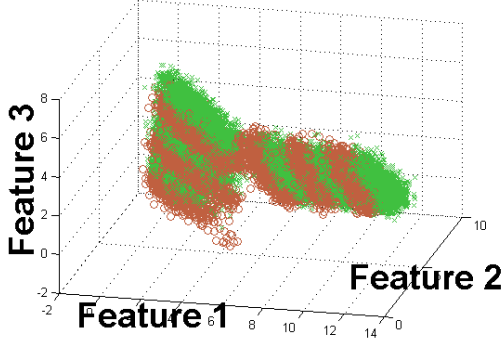


other, the diagonal Gaussian observation model is still valid. We use the same training and testing split as the “clean” experiment and learn two HMMs (recognizers), which we refer to as “swapped.” In the “swapped” setting, highly informative feature(s) still exist just as in the “clean” setting; however, traditional HMMs with MLE training are unable to identify this feature. The disturbance of the irrelevant dynamics creates an inferior model, which yields 10.1% error in testing.

Next, we apply the SBHMMs to the “swapped” data. The piecewise i.i.d. assumption allows the SBHMMs to select different features for different temporal segments (states) obtained by the initial round of the *Viterbi* decoding. Weight α_j of boosting can be considered as an indicator of “importance” of the corresponding feature j . Figure 9(a), the feature weights of the two SBHMMs, shows that segmental feature selection successfully identifies feature 3 as the informative feature in the first three states, feature 1 as that in the last three states, and feature 2 as the uninformative feature all the time. The SBHMMs then project the data into the 12-dimensional output space. To illustrate the distribution after this nonlinear projection in low dimension, we compute the most discriminatory $W_{12 \times 1}$

Table 16: Synthetic example (segmental)

model	feature	state 1	state 2	state 3	state 4	state 5	state 6
HMM ₁ (orange)	f1	1	2	3	7	9	11
	f2	6	5	4	4	5	6
	f3	1	3	5	3	2	1
HMM ₂ (green)	f1	1	2	3	8	10	12
	f2	6	5	4	4	5	6
	f3	2	4	6	3	2	1



projection using Fisher Linear Discriminant Analysis (FLDA) [23]. The data are plotted in Figure 9(b), which the Y-axis being the projected feature value and X-axis being the data index (2,000 sequences times six sample points per sequence). Although they are still not completely separated, the HMMs in the segmentally-boosted new feature space are capable of using the temporal correlation to disambiguate the two classes. SBHMMs achieve an error rate of 2.3%; the reduction of error is 77.2%.

This set of experiments shows that the segmental feature selection can extract discriminative features even if they are just “sometimes informative” while traditional HMMs cannot. In the following sections, we show that in many real world applications in which one discriminative feature for class separation is unlikely, SBHMMs join the efforts of different features at different times in order to reduce error in recognition.

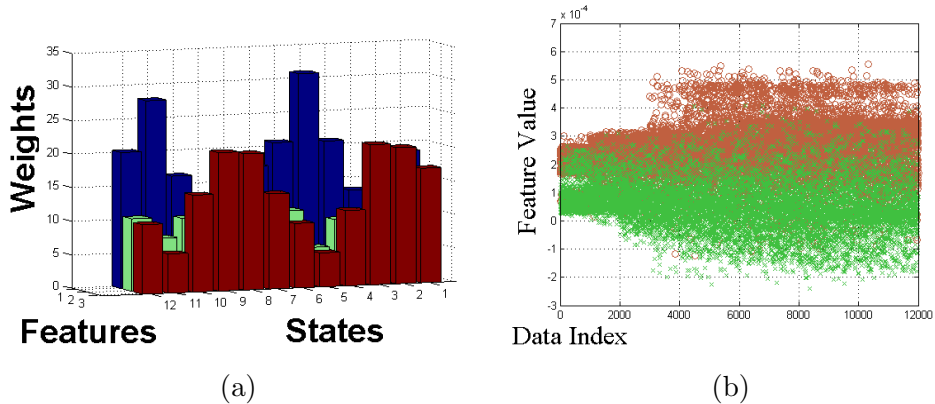


Figure 9: (a) feature weights obtained from the SBHMMs; (b) the data after segmental feature selection, projected in 2D by FLDA

6.2 American Sign Language Recognition Results

In the application of continuous American Sign Language recognition (ASLR), we compare SBHMMs with two baseline HMMs [93, 61]¹. In the first continuous recognition experiment, 500 sentences of 40 different signs are performed by one subject in five-sign phrases. We compute 16 features from the two hand blobs, including the position, the velocity, the size, the length and the angle of the first eigenvector, and the eccentricity of bounding ellipse from a color-based hand tracker. We choose the same 400 sentences for training and 100 sentences for testing as the original researchers [93]. We use four-state HMMs, shown in Figure 10(a), for recognition. The second continuous recognition experiment uses the “Acceleglove” [102], which has two-axis accelerometers mounted on a glove, an elbow, and a shoulder providing 17 features, such as wrist rotation and hand movement [102]. This dataset contains 665 sentences with 141 different signs. We use three-state HMMs, shown in Figure 10(b), on this dataset with ten-fold cross-validation. The experimental results are listed in Table 18. Despite the high accuracy of the original HMM baselines, SBHMMs are able to reduce the error rate by about 20% on both datasets, with or without postprocessing by grammar, as shown in Table 18.

To prove that SBHMMs actually select the right features, we manually examined the selected features of the Acceleglove dataset. The meanings of the 17 features, including 15

¹Both datasets are available at <http://wiki.cc.gatech.edu/ccg/projects/asl/asl>.

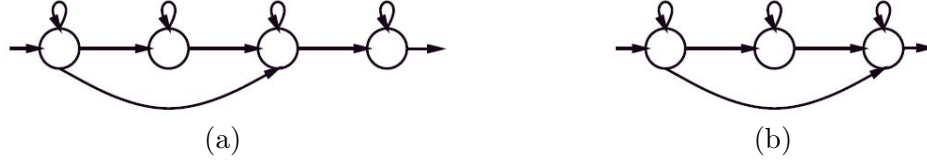


Figure 10: (a) HMM used in vision-based ASLR. It contains four states, with one “skip” link (b) HMM used in Accleglove-based ASLR. It contains three states, with one “skip” link.

accelerometers (the first 15) and two potentiometers (the last two) are listed in Table 17. We found that SBHMMs corrected three instances of misclassification made by HMMs on the minimal pair BROTHER and SISTER.² The signs BROTHER and SISTER are illustrated in Figure 11. The only difference between the two signs is the initial location of the hand, which should relate to the readings by the accelerometers on the shoulder. Indeed, the feature weight computed by the SBHMMs in Figures 12(a) and (c) shows that the readings of the accelerometers on the shoulder are considered moderately important when we use all 141 different signs for training.

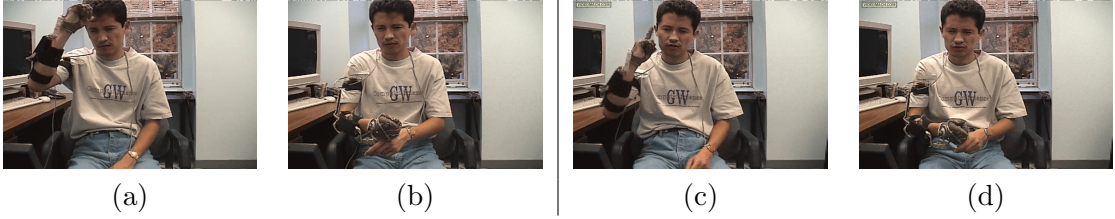
To show the impact of the minimal pairs in feature weighting, we also conducted training using only the samples of BROTHER and SISTER, and then using all the signs including SISTER but excluding BROTHER. The feature weights are plotted in Figure 12(b), (d), and (e). Note that the red dots are the weight assigned to the first state. We can see that when using only the minimal pair of BROTHER and SISTER to train, the weight of the shoulder accelerometers at the beginning (first state) is significantly higher in Figure 12(b) and (d), which illustrate the power of segmental feature selection. In addition, when training with the 140 types of signs without BROTHER, the weight of the shoulder accelerometer for SISTER is reduced in Figure 12(e) because of the relaxed competition in classification. Note that other signs that need the reading of the accelerometers on the shoulder to discriminate may still exist. Therefore, their weights do not necessarily vanish ³.

²The minimal pair is defined in Appendix A. In proper ASL, BROTHER and SISTER are disambiguated by both the starting position and the initial handshape. However, in this dataset, collected previously using a non-native signer, the signing of the two are effectively minimal pairs with the only difference being the starting position.

³Some change in the weight of irrelevant features can be a result of overfitting because SISTER and BROTHER share sign parts (inseparable states), as explained in Chapter 4 and Section 6.5.

Table 17: The meaning of the 17 accelerometer readings

feature index	meaning
1 and 2	thumb outside
3 and 4	thumb top (on thumbnail)
5 and 6	index finger
7 and 8	middle finger
9 and 10	ring finger
11 and 12	little finger
13	wrist perpendicular to bones
14	wrist parallel to fingers
15	wrist perpendicular to palm
16	shoulder elevation (forward)
17	shoulder (outward)

**Figure 11:** Illustration of the formation of the minimal pair BROTHER and SISTER. (a)(b) BROTHER and (c)(d) SISTER.

In summary, the validation on ASLR data shows that SBHMMs assign heavy weights on features such as shoulder elevation, which is considered meaningful by our sign language expert, Dr. Harley Hamilton.

6.3 Human Gait Identification Results

We compare SBHMMs with the human gait recognition results previously reported by Kim and Pavlovic [47]. In their paper, the performance of several established discriminative training methods for mixtures of Bayesian network classifiers such as conditional maximum

Table 18: Comparison of the test error on vision-based ASLR (top) and accelerometer-based ASLR (bottom)

error	With grammar			Without grammar		
	HMM	SBHMM	reduction	HMM	SBHMM	reduction
vision	2.2%	1.4%	36.4%	3.2%	2.0%	37.5%
accel.	2.2%	1.8%	17.1%	4.9%	3.8%	22.4%

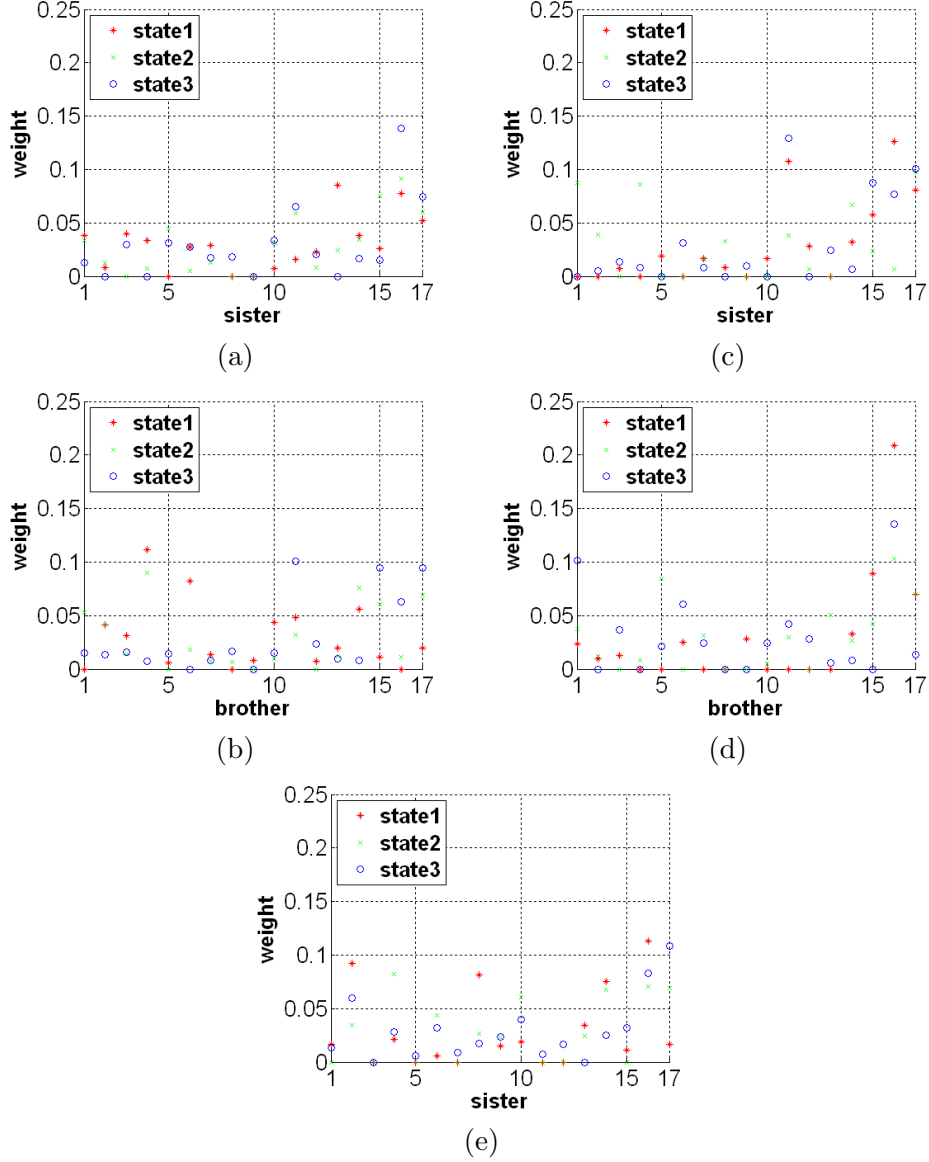


Figure 12: The impact of minimal pairs in feature weighting obtained by the SBHMMs. (a) and (b): the feature weighting for the signs SISTER and BROTHER, trained with all the classes; (c) and (d): the feature weighting for the signs SISTER and BROTHER, trained with only those two signs; (e): the feature weighting for the sign SISTER, trained with all the classes except BROTHER.

Table 19: Comparison of the test error on Georgia Tech speed-control gait dataset. The first five rows are directly from Kim and Pavlovic [47]

Approach	Error
1-NN DTW	8.38±3.68%
HMM	11.50±4.78%
BML [39]	10.13±3.61%
MixCML [47]	4.00±3.48%
BoostML [74]	11.87±5.11%
BHMM [110]	5.93±6.64%
SBHMM	3.44±1.43%

likelihood (CML) are evaluated using the gait data [96]. This dataset⁴ consists of nine trials of 15 subjects walking at four different speeds. The data record the 3D position of 22 markers on the subject at 120Hz. Following the authors’ convention exactly, we obtain 180 sub-sequences for the five subjects at various speeds. These sequences are then truncated to sub-sequences of length around 80. Each sequence contains six dimensional feature vector describing the joint angle of torso-femur, femur-tibia, and tibia-foot. We randomly choose 100 sequences for training and the other 80 for testing. We execute such random training/testing splits ten times and report the average test error in Table 19, in which three-state HMMs with a single Gaussian are used to identify the subjects. This table shows that the SBHMM outperforms all the other six algorithms, and the reduction in the mean test error ranges from 14% (MixCML [47]) to 70% (HMM). The variance of the error with SBHMMs is also relatively low compared to the other methods.

6.4 Audio and Visual Speech Recognition Results

We also test our SBHMM algorithm on the Georgia Tech Speech Reading (GTSR) dataset [110] with two tasks: lip reading [76] (visual feature only) and speech recognition (acoustic feature only)⁵. The visual features are 18 infrared trackers around the lip. Their three-dimensional positions are recorded at 120Hz. The audio features are the first 13 orders of MFCCs [80] and their derivatives, computed from a 10ms sliding window at 120Hz from a 16kHz sound track. The ground truth segmentation is obtained by forced alignment using CMU Sphinx [53].

⁴The dataset is available at ftp://ftp.cc.gatech.edu/pub/gvu/cpl/walkers/speed_control_data/.

⁵The dataset is available at <http://www.cc.gatech.edu/cpl/projects/speechreading/index.html>.

Table 20: Georgia Tech Speech Reading Database

Total Length	30m45s	Sampling Rate	120Hz
Training Data	24m42s	Testing Data	06m03s
Total Sentences	275	Total Phones	8468
Total Phonemes	39	Total Samples	>200,000

Table 21: Comparison of the test error on visual lip reading (top) and acoustic speech recognition (bottom).

	HMM	BHMM [110]	SBHMM	AdaBoost
Visual	50.36±1.16%	42.56±1.11%	34.16±1.85%	60.18±0.00%
Acoustic	32.30±2.06%	26.54±0.83%	19.65±1.00%	39.69±0.00%

Since the goal is to illustrate automatic feature selection on the baseline HMM, we did not perform elaborate preprocessing steps as most state-of-the-art speech recognizers do. The dataset is described by Table 20. Both visual and acoustic recognition systems are implemented using three-state HMMs with diagonal Gaussians mixtures. Note that the visual phonemes (visemes) are defined the same as the acoustic phonemes, and the reported test errors are averaged from five runs. SBHMMs reduce the test error by 30% compared to HMMs in Table 21. Tables 19 and 21 also illustrate that, by assuming piecewise i.i.d. instead of i.i.d. for the entire sequence, the SBHMM has higher accuracy than the boosted HMM (BHMM) [110], which selects features using $(x = o_t, y = c)$. The “AdaBoost” results in Table 21 selects feature using $(x = o_t, y = c)$ without temporal smoothing through HMMs.

Table 22: The two highest generalized Rayleigh quotients

$\frac{w^T S_B w}{w^T S_W w}$	HMM	SBHMM
generalized eigenvalue 1	1.9004	10.0361
generalized eigenvalue 2	0.8140	1.8098

Table 23: Average likelihood ratio of the correct over maximal incorrect HMM decoding

	HMM	SBHMM
Ratio	0.87	1.16

We further validate our approach by comparing SBHMMs with traditional HMMs on the GTSR dataset based on four criteria as follows:

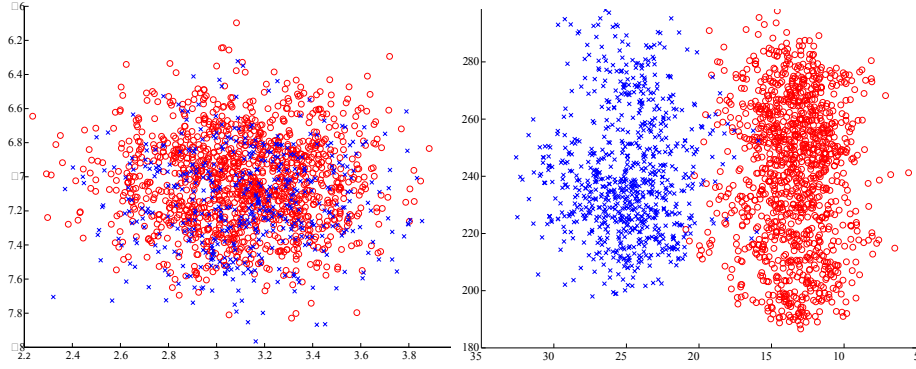


Figure 13: Sample distribution of viseme /EY/ and /N/ in the feature space of the standard HMM (left) and in the feature space computed by the SBHMMs, both then re-projected to the most discriminatory 2D space by FLDA.

Table 24: Average variance of the diagonal Gaussians of the HMM

	HMM	SBHMM
Avrg Var	0.2573	0.0136

- Class Separability

We compute the generalized Rayleigh quotient, which is the ratio of the interclass variance to the intraclass variance, illustrated in Table 22. The higher generalized eigenvalue of the quotient reflects the better separability of the data. SBHMMs manage to obtain a generalized first eigenvalue an order of magnitude higher than traditional HMMs, which illustrates the effectiveness of the discriminative feature space. As a qualitative study, we also randomly picked two viseme (the visual phoneme [14]) classes /EY/ and /N/ and plot their sample distribution in the feature space of HMMs and SBHMMs (re-projected to two dimensions using FLDA). As can be seen, the class separability is improved by segmental feature selection and the non-linear projection.

- Model Discrimination

In Equation 2 in Chapter 3, the model distance is the average likelihood ratio of the correct label to the incorrect label, which is usually differentiable and convenient for optimization. In the classification stage, the label is actually determined by Equation 1 in Chapter 3. Therefore, a high likelihood ratio of $\frac{P(\mathbf{O}^c|\lambda_c)}{\max_{v \neq c} P(\mathbf{O}^c|\lambda_v)}$ (discriminative power) is desirable. Table 23 shows that the average “correct:incorrect”

likelihood ratio for the standard HMMs in the GTSR data is lower than 1.0, but this ratio is remarkably improved by the SBHMMs. We are able to achieve better model discrimination, that is, to make more confident and accurate predictions, after segmental discriminative feature selection.

- Model Precision

For statistical reasons (*e.g.*, maximum entropy) and technical convenience, the Gaussian observation model is widely used for HMMs. In practice, the unknown underlying data distribution may or may not be Gaussian. Therefore, the mixture of the Gaussian (MoG) model is introduced to compensate for the discrepancy. However, the number of Gaussians in the mixture model is yet another parameter to be tuned. If each feature is *normalized* within a range of $[-1, 1]$, we consider the width of the Gaussians as an indicator of the modeling precision: a wide Gaussian indicates that the samples are sparsely distributed and that such a high variance may result from inaccurate modeling (of using one Gaussian), while a narrow Gaussian indicates that the samples are compactly distributed around the mean, and produces a “thin” enclosure of HMM sample points. We obviously prefer a “thin” enclosure to “fat” ones for classification. Table 24 shows that the diagonal covariance of the SBHMMs is much lower than that of the HMMs, thanks to the discriminative feature space by segmental feature selection.

- Model Fitness

We apply statistical tests to show that segmental feature selection for HMM increases the “Gaussianness” of the data in the discriminative feature space, which in turn improves the quality of the learned Gaussian observation model. It also sheds some light on the improvement of the model precision mentioned above.

No convenient statistical procedures are available to test a multidimensional Gaussian assumption, but all marginal distributions have to be univariate Gaussian if the joint distribution is a multivariate Gaussian. Therefore, we test the Gaussianness

of the marginal distribution (along each feature dimension) using the normal probability plot, the skewness measure, the unbiased kurtosis measure, z-tests, and the Kolmogorov-Smirnov test. (Mathematical definitions and details are in Appendix B.)

The normal probability plot qualitatively illustrates how the real data distribution differs from the ideal Gaussian in the accumulated distribution. The dashed straight line in Figures 14(a)-(b) is the accumulated distribution of an ideal Gaussian. If the real data are from a distribution similar to an ideal Gaussian, the “dots” are expected to remain close to the line. Figure 14(a) indicate that the marginal distribution of the original data has a big tail, which is quite different from the Gaussian distribution; the sample distribution produced by the nonlinear projection of SBHMMs in Figure 14(b) is much closer to a Gaussian distribution.

For the four quantitative statistical tests, a true Gaussian distribution will have a score of 0; and the lower the score, the closer the distribution to a true Gaussian. Figure 14(c) compares the statistical test results averaged⁶ over all the dimensions of all the viseme classes, and Figure 14(d) compares the maximum value of a single dimension (the “worst dimension”) of the sample distribution of HMMs and SBHMMs. It again validates the choice of using the output space of segmental boosting as the new feature space.

6.5 Results with Discriminative State-Space Clustering

This section presents results on DISC. We first validate DISC using the accelerometer-based ASLR dataset [61]. We choose this dataset because its 665 sentences contain 141 signs, allowing more opportunities for state-tying than the vision-based dataset of 40 signs. Without state-space clustering, three-state SBHMMs create a $3 \times 141 = 423$ set of features. We expect DISC to find a more parsimonious model representation that results in comparable accuracy.

⁶When we compute this average, the positive and negative values will cancel each other out. To avoid cancellation, we take the sum of the absolute values. Since the sign of the kurtosis matters, we examine the sign of the individual dimensions of the kurtosis test, most of which are positive (leptokurtic).

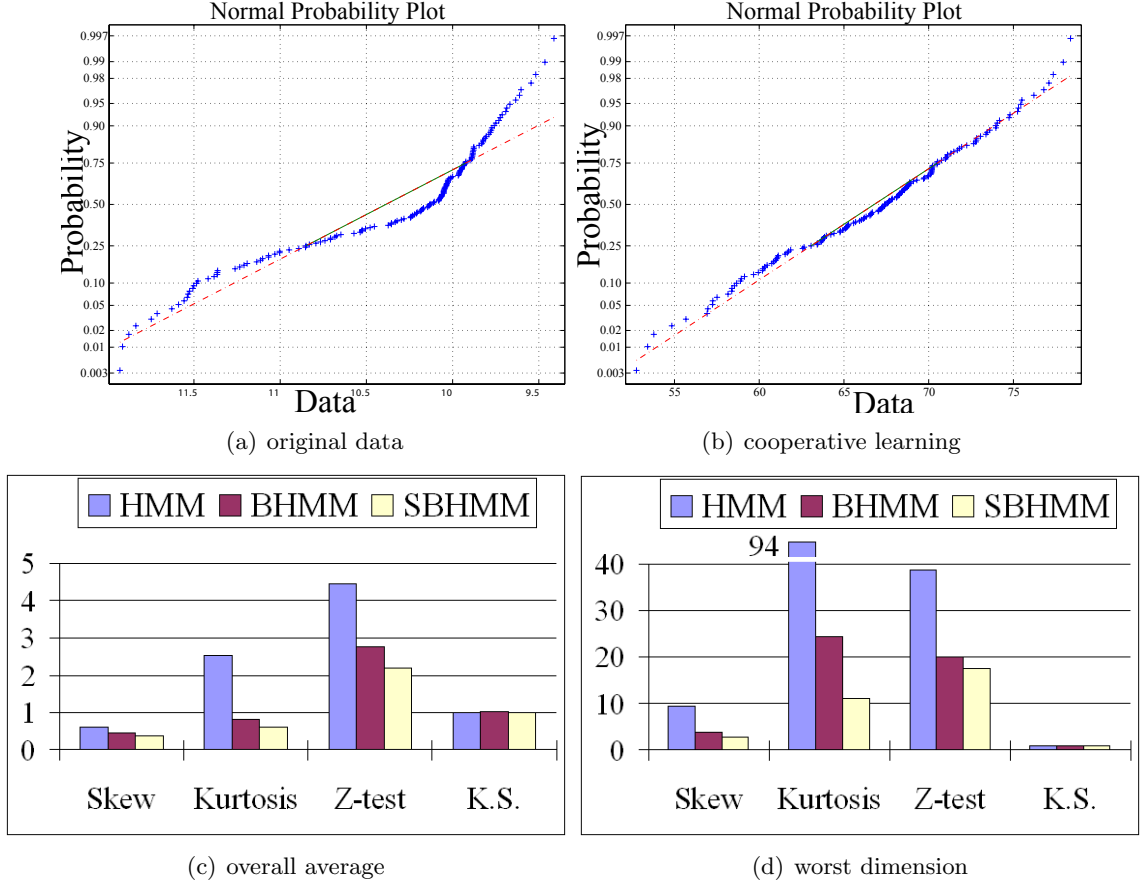


Figure 14: Statistical measurements

Table 25 shows the top ten confusable state-pairs in one run of the ten-fold validation and the SBHMMs reported in Section 6.2. The rightmost column of Table 25 illustrates the meaning or reason of the top ten confusable state-pairs, which are also candidates for sub-sign units in our approach, to be merged. For example, WIFE/HUSBAND is composed of FEMALE/MALE, a transitional move, and MARRIAGE respectively. DISC successfully identified the shared state of MARRIAGE. Another example is APOLOGIZE, which contains a repetitive movement of hand cycling in front of the chest. DISC suggested that such repetitive states can be merged and thus reduced the 3-state HMM of APOLOGIZE to a 2-state HMM. After tying the top ten confusable state-pairs and re-training the SBHMMs with less states, we achieved the recognition results by the same ten-fold cross-validation data shown in Table 26. DISC-SBHMMs reduce the error of HMMs by 29%, and reduce the error of SBHMMs by 13%. In this study, DISC-SBHMMs, which have a more compact

Table 25: Top ten confusable states computed by SBHMMs on the third fold of the accelerometer-based ASLR dataset

Rank	Sign Name	State Number	Sign Name	State Number	Semantic Reason
01	DAUGHTER	3	SON	3	BABY
02	WIFE	3	HUSBAND	3	MARRIAGE
03	MORNING	1	THING	1	stretching arm
04	APOLOGIZE	1	APOLOGIZE	2	repetitive pattern
05	WATER	1	WINE	1	W
06	ROOSTER	1	ROOSTER	2	repetitive pattern
07	TOILET	1	TOILET	2	repetitive pattern
08	SICK	1	SMART	2	folding middle finger
09	SUSPECT	2	PUZZLE	2	touch forehead
10	USE	1	USE	2	repetitive pattern

model, achieve comparable accuracies with SBHMMs; we believe that the reduction of error is achieved by less overfitting in SBHMMs after state tying.

Table 26: Results of DISC-SBHMMs. Comparison of the test error on the accelerometer-based ASLR dataset based on ten-fold cross-validation.

HMMs	SBHMMs	DISC-SBHMMs
$4.85 \pm 1.78\%$	$3.83 \pm 1.24\%$	$3.29 \pm 1.13\%$

Further investigating the impact on recognition accuracy by SBHMMs and DISC-SBHMMs, Table 27 lists paired t-tests (mathematical definition and details in Appendix B) on the ten-fold Accelglove-ASLR data. All comparisons are statistically significant. In other words, DISC-SBHMMs and SBHMMs consistently improve traditional HMM result in the tests.

Table 27: Paired t-test results of testing error using HMMs, SBHMMs, and DISC-SBHMMs on the accelerometer-based ASLR dataset based on ten-fold cross-validation.

	SBHMMs versus HMMs	DISC-SBHMMs versus HMMs	DISC-SBHMMs versus SBHMMs
t-value	2.67	3.44	2.06
one-sided p-value	$1.29 * 10^{-2}$	$3.69 * 10^{-3}$	$3.46 * 10^{-2}$

In order to test the consistency of DISC-SBHMMs, we manually examined the ten most confusable pairs for each of the ten folds in cross-validation. These 100 most confusing state pairs reported by DISC-SBHMMs contains 35 distinct pairs: 20 state pairs are reported once and the other 15 state pairs are discovered by more than one fold (80 pairs). On the one hand, the high overlap in most confusing pairs reported by different folds illustrates

Table 28: Top six confusable state clusters (from top ten pairs) computed by SBHMMs for one subject of the CopyCat dataset.

Rank	Sign Names and State Numbers	Semantic Reason
01	SNAKE state 1 and WHITE state 1	hand position
02	ORANGE state 2 and ORANGE state 3	repetitive pattern
03	SNAKE state 2, WHITE state 2, FLOWERS state 1, and FLOWERS state 4	hand position
04	BLUE state 1, BLUE state 3, and BLUE state 4	repetitive pattern
05	BOX state 3, BOX state 4, and CHAIR state 4	hand location
06	WALL state 1, WALL state 2, and WALL state 3	repetitive pattern

that the tied states are not obtained by chance. It proves that a consistent tying can be inferred from data automatically. On the other hand, the other distinct pairs suggests that more interesting patterns may be extracted with more data. From these results, we believe that DISC-SBHMMs can extract a meaningful set of sub-sign units, which provide a parsimonious and discriminative representation for ASLR.

Then, we run DISC on the CopyCat dataset which consists of multiple children signing three-, four-, and five-sign phrases. Due to the high signing variations compared to the features and samples available, a direct application of the DISC algorithm produces much noise, although DISC is still able to identify and tie states of the only minimal pair BLUE and GREEN in the data. In order to reduce the variations among different children, we picked the subject with the clearest sign and applied DISC to her signing only. Although noise, such as matching of transitional moves and misalignments, still exists, DISC on one subject produces some meaningful tying results, listed in Table 28.

6.6 Probes

ASL verification requires a rejection threshold for the similarity measurement (the likelihood in our application). Usually, a rejection threshold is empirically optimized using a hold-out validation set, then any sign that has lower likelihood in testing will be rejected (as is the input phrase containing it). In this experiment, we compare the rejection threshold computed manually using exhaustive search and the proposed automatic probe method. In one implementation, we set the probe threshold in the following way: we first train the standard

HMM model and then compute the likelihood of the forced alignment of the temporally reversed input sequences to the groundtruth scripts. For example, if the original signing is ALLIGATOR ON WALL described by frame sequence $F = \langle f_1, f_2, \dots, f_{T-1}, f_T \rangle$ in time T , we compute the likelihood for $F' = \langle f_T, f_{T-1}, \dots, f_2, f_1 \rangle$.⁷ F' is a “garbage” input for the trained model, and the likelihood of $L(\text{ALLIGATOR}, F'_{\text{ROTAGILLA}})$, $L(\text{ON}, F'_{\text{NO}})$, and $L(\text{WALL}, F'_{\text{LLAW}})$ will be used as the threshold to reject incorrect signing. We choose the temporally reversed input as a probe for many reasons: it is convenient to acquire, simple to compare (the same length as the true input), and reasonably similar to the true input (a reversed sequence still contains some form of structure, compared to a completely random sequence). In addition, the reversed input can easily adapt to the global parameters of a particular child (head/hand default position), and it has the same means/distributions of movements and location features.

Tables 29 and 30 compare the verification results in CopyCat using a manual exhaustive search of the optimal rejection threshold (tied for all signs) and automatic computation by the probe; Figures 15 and 16 summarize the same results by a receiver operating characteristic (ROC) curve. In the ROC curve, good results should be as close to the top left corner (100% true positives and 0% false positives) as possible. The automatic computed probe achieves results comparable to the best results by a brute force search in CopyCat. In CopyCat-Adult, the automatic threshold adaptation by a probe at the sign level actually achieves slightly superior performance over the manual search. Note that the methodology of our comparison here actually favors manual search, which actually does not have the access to a ROC curve that is computed using the test data. In practice, an “optimal” threshold in the ROC curve from a separate validation set may be sub-optimal for the test set, and thus yields even inferior performance than reported here. In contrast, the probe does not need a separate validation set, so the probe results in this dissertation are the “true” performance.

⁷Due to disfluencies in the signing, the input sequences are actually used in a backward order. ALLIGATOR ON WALL will be LLAW NO ROTAGILLA. We adopted this convention in our training and testing: that is, we actually compute the probe using forward input on models trained with backward data. In this dissertation, however, we still present the sequences in forward order to simplify the discussion.

Because verification is rather a subjective task (*e.g.*, whether a sloppy signing should be considered match or no-match), the criterion for human eyes and machines can be quite different. Generally speaking, the performance of the automatic verifier, even with probe, is still less accurate than the human verifier we have. The main problem is high false positives. One may use a more strict rejection threshold, but it will cost more in false negatives in the current system. After carefully examining the failed cases of the probe method, we found that some error made by the probe method is due to the simple experimental setup: certain signs, such as ALLIGATOR, are composed by repetitive movements over time, so the temporally-reversed inputs look very similar to the original inputs and thus create false negatives. This problem can be fixed by alternative designs described next.

The currently used probe is tailored to the needs of the CopyCat game with an assumption that false positives and false negatives are weighted equally. Other applications may weigh these two errors differently. For example, false positives are undesired in an educational verification task, such as CopyCat. Therefore, we can use minimal pairs as the probe input. For example, when the verifier expects MOTHER NEED BLUE PAPER, the probe can be FATHER MUST GREEN CHEESE, because (MOTHER, FATHER), (NEED, MUST), (BLUE, GREEN), and (PAPER, CHEESE) are all minimal pairs. The user that is signing will need to be very precise to be accepted as correct. In contrast, false negatives are rather annoying in an entertainment game that requires user engagement, such as CopyCat. We can use fully random input as the probe, which represents a very loose threshold. Moreover, when a sign exists in multiple phrases, the reject thresholds can be tied to the minimum value to emphasize recall rate (reducing false negatives).

6.7 Phrase Selection

This experiment compares different error ranking prediction methods. Denote the groundtruth error ranking of testing phrases, which is unreachable in the real world as GTH, a randomly-generated ranking prediction that serves as a control group as RAN, the ranking of training phrases, which is arguably “the best you can do” as TAT, and finally the proposed bigram ranking predictor as BIG. We measure the average Spearman footrule distance aDL_1 of

Table 29: CopyCat verification results with manually specified thresholds and the automatically computed threshold by the probe

Threshold	True Positive rate (%)	False Positive Rate (%)	Accuracy (%)
-120	72.0	32.1	70.5
-160	83.7	59.7	76.6
-200	91.7	60.5	76.4
-240	98.0	73.4	77.1
-320	98.0	73.4	77.1
-400	99.3	83.1	75.2
$-\infty$	100	100	71.4
Probe	98.4	73.6	77.1

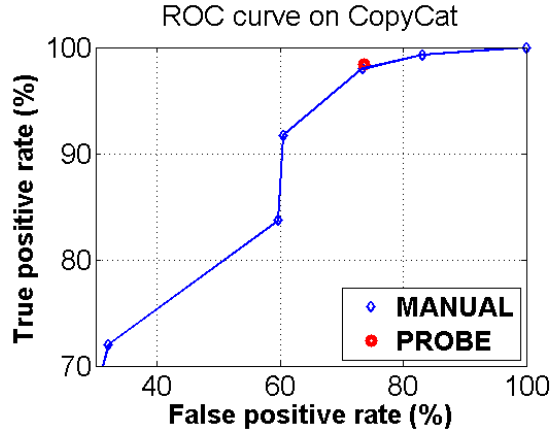


Figure 15: ROC curve of the verification results in CopyCat.

Table 30: CopyCat-Adult verification results with manually specified thresholds and the automatically computed threshold by the probe

Threshold	True Positive rate (%)	False Positive Rate (%)	Accuracy (%)
-100	75.8	9.1	80.4
-110	91.9	13.6	90.2
-120	95.0	22.7	89.5
-160	97.0	38.6	86.0
-200	98.0	63.6	79.0
-240	98.0	81.8	73.4
$-\infty$	100	100	69.2
Probe	97.0	15.9	93.0

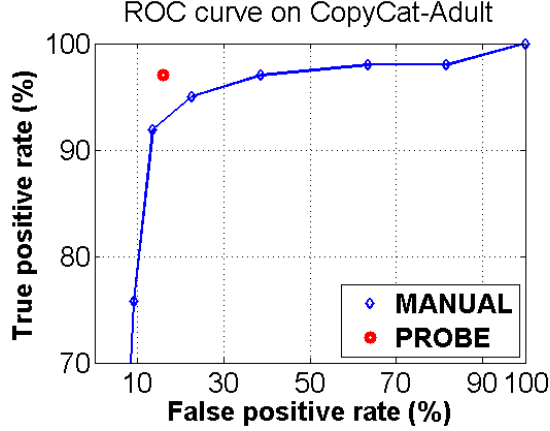


Figure 16: ROC curve of the verification results in CopyCat-Adult.

these rankings using the CopyCat data and the CopyCat-Adult data, the goal of which is to be as close to GTH as possible.

We computed the training and testing errors of the CopyCat data and the CopyCat-Adult data, ranking the phrases by their error rates in Tables 31, 33, 32, and 34, respectively. These tables are the actual TAT (training) and GTH (testing) results. From the experimental results summarized in Tables 35, 36, Figures 17, and 18,⁸ we are able to draw the following conclusions:⁹

1. BIG and TAT predictions are significantly better than RAN with a confidence greater than 98%.
2. BIG and TAT results are comparable with each other.
3. BIG uses significantly fewer resources for prediction than TAT.

These conclusions are also supported by normalized Kendall tau distance [45] and Pearson product-moment correlation coefficient (PMCC) [101], listed in Tables 37 and 38. A “good” ranking algorithm should produce a normalized Kendall tau distance close to zero and a PMCC close to one.

⁸We compute the expectation of $aDL_1(RAN, GTH)$, which has a close form solution, analytically, but we estimate the standard deviation, which is very difficult to compute, by one million simulations.

⁹Please refer to Appendix B for a brief explanation of hypothesis testing.

Table 31: The training error ranking of the CopyCat data

Phrase	Error Rate (%)	Error Rank
SNAKE BEHIND WALL	0	1
SNAKE UNDER CHAIR	14.29	2
SNAKE ON BOX	22.22	3
ALLIGATOR BEHIND WALL	33.33	4
SPIDER UNDER WAGON	33.33	4
SPIDER ON WALL	40.00	6
SPIDER UNDER BED	40.00	6
SNAKE IN FLOWERS	40.00	6
WHITE CAT ON ORANGE WALL	42.86	9
ALLIGATOR UNDER GREEN BED	42.86	9
ALLIGATOR BEHIND CHAIR	50.00	11
CAT UNDER CHAIR	50.00	11
ALLIGATOR ON BED	50.00	11
CAT ON WALL	50.00	11
ALLIGATOR ON BLUE WALL	50.00	11
ALLIGATOR IN WAGON	50.00	11
BLACK ALLIGATOR BEHIND ORANGE WAGON	50.00	11
BLACK CAT ON GREEN BED	53.85	18
GREEN SPIDER UNDER ORANGE CHAIR	58.33	19
SPIDER ON CHAIR	60.00	20
ALLIGATOR BEHIND BLACK WALL	60.00	20
WHITE CAT IN GREEN BOX	60.00	20
SNAKE UNDER BLACK CHAIR	60.00	20
SNAKE IN GREEN WAGON	60.00	20
ORANGE SPIDER UNDER GREEN FLOWERS	61.54	25
BLACK CAT IN BLUE WAGON	61.54	25
SPIDER UNDER BLUE CHAIR	62.50	27
WHITE ALLIGATOR ON BLUE WALL	63.63	28
SPIDER IN BOX	66.67	29
CAT BEHIND BOX	66.67	29
CAT BEHIND BED	66.67	29
CAT BEHIND ORANGE BED	66.67	29
SPIDER IN ORANGE FLOWERS	66.67	29
ALLIGATOR IN ORANGE FLOWERS	66.67	29
SNAKE UNDER BLUE CHAIR	66.67	29
BLACK SNAKE UNDER BLUE CHAIR	66.67	29
SNAKE UNDER BLUE FLOWERS	66.67	29
ORANGE SPIDER IN GREEN BOX	66.67	29
BLACK CAT BEHIND GREEN BED	66.67	29
SPIDER ON WHITE WALL	70.00	40
ALLIGATOR IN BOX	71.43	41
ORANGE ALLIGATOR IN GREEN FLOWERS	71.43	41

Table 31 continued.

Phrase	Error Rate (%)	Error Rank
ALLIGATOR BEHIND BLUE WAGON	75.00	43
BLUE ALLIGATOR ON GREEN WALL	75.00	43
SNAKE UNDER BED	77.78	45
BLACK SPIDER IN WHITE FLOWERS	77.78	45
WHITE SNAKE IN BLUE FLOWERS	80.00	47
CAT UNDER ORANGE CHAIR	80.00	47
GREEN ALLIGATOR UNDER BLUE FLOWERS	80.00	47
CAT ON GREEN WALL	80.00	47
CAT UNDER BLUE BED	83.33	51
GREEN SNAKE UNDER BLUE CHAIR	85.00	52
ORANGE SNAKE UNDER BLUE FLOWERS	85.71	53
BLUE SPIDER ON GREEN BOX	85.71	53
ALLIGATOR BEHIND ORANGE WAGON	87.50	55
CAT BEHIND FLOWERS	100.00	56
SPIDER IN BLUE BOX	100.00	56
SPIDER IN GREEN BOX	100.00	56
CAT ON BLUE BED	100.00	56

Table 32: The training error ranking of the CopyCat-Adult data

Phrase	Error Rate (%)	Error Rank
SNAKE BEHIND WALL	0.00	1
SPIDER ON WALL	0.00	1
CAT ON WALL	0.00	1
SNAKE UNDER CHAIR	0.00	1
ALLIGATOR BEHIND CHAIR	0.00	1
CAT UNDER CHAIR	0.00	1
SPIDER IN BOX	0.00	1
SNAKE ON BOX	0.00	1
CAT BEHIND BED	0.00	1
ALLIGATOR ON BED	0.00	1
SNAKE UNDER BED	0.00	1
SPIDER UNDER BED	0.00	1
ALLIGATOR IN WAGON	0.00	1
SPIDER UNDER WAGON	0.00	1
SPIDER IN ORANGE FLOWERS	0.00	1
CAT ON GREEN WALL	0.00	1
ALLIGATOR UNDER GREEN BED	0.00	1
ALLIGATOR IN ORANGE FLOWERS	0.00	1
CAT BEHIND ORANGE BED	0.00	1
CAT UNDER ORANGE CHAIR	0.00	1

Table 32 continued.

Phrase	Error Rate (%)	Error Rank
SNAKE UNDER BLUE CHAIR	0.00	1
SPIDER ON CHAIR	0.00	1
SNAKE IN FLOWERS	0.00	1
SPIDER IN BLUE BOX	0.00	1
WHITE CAT IN GREEN BOX	0.00	1
BLACK ALLIGATOR BEHIND ORANGE WAGON	7.69	26
ORANGE ALLIGATOR IN GREEN FLOWERS	9.09	27
BLACK CAT ON GREEN BED	10.00	28
BLACK SNAKE UNDER BLUE CHAIR	12.50	29
WHITE CAT ON ORANGE WALL	15.00	30
ALLIGATOR IN BOX	16.67	31
BLACK SPIDER IN WHITE FLOWERS	16.67	31
BLACK CAT BEHIND GREEN BED	19.05	33
ORANGE SPIDER IN GREEN BOX	20.00	34
ALLIGATOR ON BLUE WALL	20.00	34
ALLIGATOR BEHIND ORANGE WAGON	25.00	36
SPIDER UNDER BLUE CHAIR	28.57	37
GREEN SPIDER UNDER ORANGE CHAIR	29.41	38
SNAKE IN GREEN WAGON	30.00	39
SPIDER IN GREEN BOX	33.33	40
ALLIGATOR BEHIND BLACK WALL	33.33	40
BLACK CAT IN BLUE WAGON	40.00	42
SPIDER ON WHITE WALL	40.00	42
GREEN ALLIGATOR UNDER BLUE FLOWERS	42.86	44
GREEN SNAKE UNDER BLUE CHAIR	43.48	45
ORANGE SNAKE UNDER BLUE FLOWERS	44.44	47
ALLIGATOR BEHIND WALL	50.00	48
CAT BEHIND BOX	50.00	48
CAT BEHIND FLOWERS	50.00	48
ALLIGATOR BEHIND BLUE WAGON	50.00	48
SNAKE UNDER BLUE FLOWERS	50.00	48
WHITE ALLIGATOR ON BLUE WALL	50.00	48
WHITE SNAKE IN BLUE FLOWERS	55.56	54
ORANGE SPIDER UNDER GREEN FLOWERS	55.56	54
BLUE SPIDER ON GREEN BOX	66.67	56
SNAKE UNDER BLACK CHAIR	80.00	57
BLUE ALLIGATOR ON GREEN WALL	81.82	58
CAT ON BLUE BED	100.00	59

Table 33: The true error ranking of the CopyCat data

Phrase	Error Rate (%)	Error Rank
SNAKE UNDER CHAIR	0.00	1
ALLIGATOR BEHIND WALL	0.00	1
SPIDER UNDER WAGON	0.00	1
ALLIGATOR BEHIND CHAIR	0.00	1
CAT UNDER CHAIR	0.00	1
ALLIGATOR ON BED	0.00	1
CAT ON WALL	0.00	1
ALLIGATOR ON BLUE WALL	0.00	1
SPIDER ON CHAIR	0.00	1
SPIDER IN BOX	0.00	1
SPIDER ON WHITE WALL	0.00	1
ALLIGATOR BEHIND BLUE WAGON	0.00	1
SNAKE UNDER BED	0.00	1
CAT UNDER BLUE BED	0.00	1
ORANGE SNAKE UNDER BLUE FLOWERS	0.00	1
CAT BEHIND FLOWERS	0.00	1
SPIDER ON WALL	33.33	17
SPIDER UNDER BED	33.33	17
CAT BEHIND BOX	33.33	17
CAT BEHIND BED	33.33	17
CAT BEHIND ORANGE BED	33.33	17
ALLIGATOR IN BOX	33.33	17
WHITE CAT ON ORANGE WALL	36.36	23
SNAKE ON BOX	50.00	24
ALLIGATOR IN WAGON	50.00	24
ALLIGATOR BEHIND BLACK WALL	50.00	24
WHITE ALLIGATOR ON BLUE WALL	50.00	24
SPIDER IN ORANGE FLOWERS	50.00	24
ALLIGATOR IN ORANGE FLOWERS	60.00	29
SNAKE BEHIND WALL	66.67	30
BLACK ALLIGATOR BEHIND ORANGE WAGON	66.67	30
SNAKE UNDER BLUE CHAIR	66.67	30
GREEN SPIDER UNDER ORANGE CHAIR	75.00	33
WHITE SNAKE IN BLUE FLOWERS	75.00	33
GREEN SNAKE UNDER BLUE CHAIR	78.95	35
BLACK SPIDER IN WHITE FLOWERS	80.00	36
CAT UNDER ORANGE CHAIR	80.00	36
BLACK SNAKE UNDER BLUE CHAIR	83.33	38
GREEN ALLIGATOR UNDER BLUE FLOWERS	85.71	39
BLACK CAT ON GREEN BED	90.91	40
ORANGE ALLIGATOR IN GREEN FLOWERS	90.91	40

Table 33 continued.

Phrase	Error Rate (%)	Error Rank
WHITE CAT IN GREEN BOX	91.67	42
SNAKE IN FLOWERS	100.00	43
ALLIGATOR UNDER GREEN BED	100.00	43
SNAKE UNDER BLACK CHAIR	100.00	43
SNAKE IN GREEN WAGON	100.00	43
ORANGE SPIDER UNDER GREEN FLOWERS	100.00	43
BLACK CAT IN BLUE WAGON	100.00	43
SPIDER UNDER BLUE CHAIR	100.00	43
SNAKE UNDER BLUE FLOWERS	100.00	43
ORANGE SPIDER IN GREEN BOX	100.00	43
BLACK CAT BEHIND GREEN BED	100.00	43
BLUE ALLIGATOR ON GREEN WALL	100.00	43
CAT ON GREEN WALL	100.00	43
BLUE SPIDER ON GREEN BOX	100.00	43
ALLIGATOR BEHIND ORANGE WAGON	100.00	43
SPIDER IN BLUE BOX	100.00	43
SPIDER IN GREEN BOX	100.00	43
CAT ON BLUE BED	100.00	43

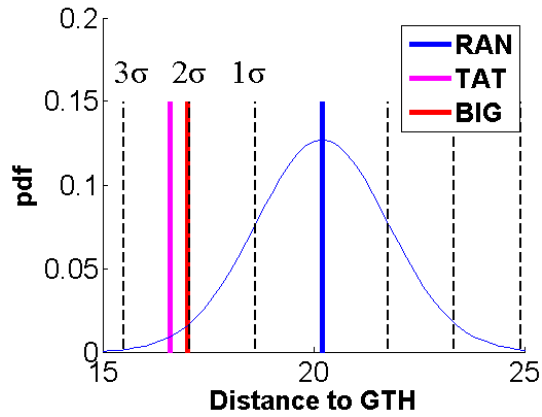


Figure 17: Ranking approximation measured by Spearman footrule distance in the Copy-Cat data.

Table 34: The true error ranking of the CopyCat-Adult data

Phrase	Error Rate (%)	Error Rank
SNAKE BEHIND WALL	0.00	1
SPIDER ON WALL	0.00	1
CAT ON WALL	0.00	1
SNAKE UNDER CHAIR	0.00	1
ALLIGATOR BEHIND CHAIR	0.00	1
CAT UNDER CHAIR	0.00	1
SPIDER IN BOX	0.00	1
SNAKE ON BOX	0.00	1
CAT BEHIND BED	0.00	1
ALLIGATOR ON BED	0.00	1
SNAKE UNDER BED	0.00	1
SPIDER UNDER BED	0.00	1
ALLIGATOR IN WAGON	0.00	1
SPIDER UNDER WAGON	0.00	1
SPIDER IN ORANGE FLOWERS	0.00	1
CAT ON GREEN WALL	0.00	1
ALLIGATOR UNDER GREEN BED	0.00	1
ALLIGATOR IN ORANGE FLOWERS	0.00	1
CAT BEHIND ORANGE BED	0.00	1
CAT UNDER ORANGE CHAIR	0.00	1
BLACK ALLIGATOR BEHIND ORANGE WAGON	0.00	1
BLACK CAT ON GREEN BED	0.00	1
ALLIGATOR IN BOX	0.00	1
BLACK SPIDER IN WHITE FLOWERS	0.00	1
ORANGE SPIDER IN GREEN BOX	0.00	1
ALLIGATOR BEHIND ORANGE WAGON	0.00	1
SPIDER UNDER BLUE CHAIR	0.00	1
SNAKE IN GREEN WAGON	0.00	1
SPIDER IN GREEN BOX	0.00	1
ALLIGATOR BEHIND BLACK WALL	0.00	1
ORANGE SNAKE UNDER BLUE FLOWERS	0.00	1
ALLIGATOR BEHIND WALL	0.00	1
CAT BEHIND BOX	0.00	1
CAT BEHIND FLOWERS	0.00	1
ALLIGATOR BEHIND BLUE WAGON	0.00	1
SNAKE UNDER BLUE FLOWERS	0.00	1
CAT ON BLUE BED	0.00	1
BLACK SNAKE UNDER BLUE CHAIR	25	38
SNAKE UNDER BLUE CHAIR	33.33	39
WHITE CAT ON ORANGE WALL	33.33	39
GREEN SPIDER UNDER ORANGE CHAIR	33.33	39

Table 34 continued.

Phrase	Error Rate (%)	Error Rank
GREEN SNAKE UNDER BLUE CHAIR	40.00	42
SPIDER ON CHAIR	50.00	43
SNAKE IN FLOWERS	50.00	43
SPIDER IN BLUE BOX	50.00	43
WHITE CAT IN GREEN BOX	50.00	43
ORANGE ALLIGATOR IN GREEN FLOWERS	50.00	43
BLACK CAT IN BLUE WAGON	50.00	43
WHITE ALLIGATOR ON BLUE WALL	50.00	43
GREEN ALLIGATOR UNDER BLUE FLOWERS	66.67	50
WHITE SNAKE IN BLUE FLOWERS	66.67	50
BLACK CAT BEHIND GREEN BED	75.00	52
BLUE ALLIGATOR ON GREEN WALL	75.00	52
ALLIGATOR ON BLUE WALL	100.00	54
SPIDER ON WHITE WALL	100.00	54
ORANGE SPIDER UNDER GREEN FLOWERS	100.00	54
BLUE SPIDER ON GREEN BOX	100.00	54
SNAKE UNDER BLACK CHAIR	100.00	54
CAT UNDER BLUE BED	100.00	54

Table 35: Spearman footrule distance computed from the CopyCat data.

$aDL_1(BIG, GTH)$	$aDL_1(TAT, GTH)$	$\mu(aDL_1(RAN, GTH))$	$\sigma(aDL_1(RAN, GTH))$
17.0	16.6	19.8	1.57

Table 36: Spearman footrule distance computed from the CopyCat-Adult data.

$aDL_1(BIG, GTH)$	$aDL_1(TAT, GTH)$	$\mu(aDL_1(RAN, GTH))$	$\sigma(aDL_1(RAN, GTH))$
12.3	11.7	18.6	1.98

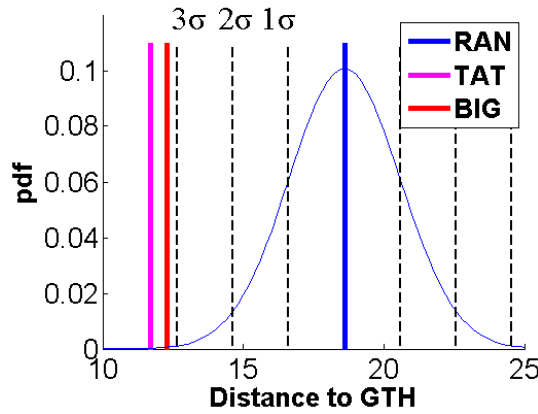
**Figure 18:** Ranking approximation measured by Spearman footrule distance in the CopyCat-Adult data.

Table 37: Kendall tau distance and Pearson product-moment correlation coefficient computed from the CopyCat data.

$K(BIG, GTH)$	$K(TAT, GTH)$	$K(RAN, GTH)$	Ideal
0.202	0.194	0.413	0.000
$\rho(BIG, GTH)$	$\rho(TAT, GTH)$	$\rho(RAN, GTH)$	Ideal
0.307	0.240	0.000	1.000

Table 38: Kendall tau distance and Pearson product-moment correlation coefficient computed from the CopyCat-Adult data.

$K(BIG, GTH)$	$K(TAT, GTH)$	$K(RAN, GTH)$	Ideal
0.412	0.326	0.440	0.000
$\rho(BIG, GTH)$	$\rho(TAT, GTH)$	$\rho(RAN, GTH)$	Ideal
0.557	0.372	0.000	1.000

6.7.1 Discussion

The results in the previous section show that BIG is comparable to TAT. On the one hand, even TAT, which requires exhaustive enumeration of all possible combination of signs, is prohibitive to compute in real applications. Thus, this finding is very promising. However, an ideal predictor should produce an average Spearman footrule distance of zero, a normalized Kendall tau distance of zero, and a PMCC of one; currently the BIG predictor is still far from satisfactory. Part of the reason is the discrepancy between training and testing data. As we can see, even the TAT prediction is not close to accurate.

6.7.2 Bi-gram versus Uni-gram

This dissertation compares a uni-gram ranking predictor (UNG) with a bi-gram ranking predictor (BIG) in Tables 39 and 40. Surprisingly, UNG accuracy is comparable to or even slightly better than BIG in CopyCat and CopyCat-Adult data. After inspecting the data, we found that this phenomenon can be mainly explained by the following two reasons.

First, the context-sensitive bi-gram model closely captures the pattern of the training data, and thus is prone to the discrepancy in training and testing data. For example, the worst BIG prediction in the CopyCat data is the phrase “CAT BEHIND FLOWERS.” BIG predicts its rank at 58, TAT predicts its rank at 56, UNG predicts its rank at 20, while GTH is at rank 1. The CopyCat dataset has a higher amount of noise, and thus the algorithm can easily overfit the training data, which has been demonstrated by other experiments.

Table 39: Comparison of uni-gram versus bi-gram models using CopyCat data

$aDL_1(BIG, GTH)$	$aDL_1(UNG, GTH)$	$aDL_1(TAT, GTH)$	$\mu(aDL_1(RAN, GTH))$
17.0	13.7	16.7	19.8

Table 40: Comparison of uni-gram versus bi-gram models using CopyCat-Adult data

$aDL_1(BIG, GTH)$	$aDL_1(UNG, GTH)$	$aDL_1(TAT, GTH)$	$\mu(aDL_1(RAN, GTH))$
12.3	13.0	12.9	12.8

Therefore, the rather simple UNG model can achieve better results. In a cleaner dataset, CopyCat-Adult, BIG works much better as shown in Table 40. The second reason is the strict grammar in the game. All words except adjectives can appear in only one place in any phrase. The restriction limits the contribution of the bi-gram estimation.

6.7.3 Suggesting New Phrases

Tables 42, 43, 44, 45, 46, and 47 list the suggested new phrases (including existing ones and unseen ones) computed by BIG and Yen’s algorithm from CopyCat and CopyCat-Adult. We set $k = 59$ to be consistent with that of the original CopyCat game. The quality of the prediction requires further validation by designing and deploying a new version of the CopyCat game. Currently, we estimate the quality of the prediction as follows. We conducted a “leave-one-out” (L1O) test, which takes 58 out of the 59 phrases in the CopyCat training data to train BIG and predict the ranking of the remaining one in the CopyCat testing data. Compared with the experiment in Table 35, BIG in this test has no information about the phrase whose ranking is to be predicted. Because the ranking of 11 phrases that contains unique bi-gram segments that cannot be predicted in our leave-one-out test, we ran a 48-fold (59-11) cross-validation on the CopyCat data. The average $aDL_1(BIG, GTH)_{L1O}$ is 13.5 as shown in Table 41, which is lower than $aDL_1(BIG, GTH)$, $aDL_1(TAT, GTH)$, and $\mu(aDL_1(RAN, GTH))$ in Table 35. Strictly speaking, $aDL_1(BIG, GTH)_{L1O}$ is averaged over 48 runs, so the number is not directly comparable to the numbers in Table 35. Nevertheless, it qualitatively illustrates that BIG is able to achieve relatively accurate ranking predictions for unseen phrases.

One additional observation is that the proposed algorithm trades signing variety for

Table 41: Result of leave-one-out (L1O) test compared with others using CopyCat data

$aDL_1(BIG, GTH)_{L1O}$	$aDL_1(BIG, GTH)$	$aDL_1(TAT, GTH)$	$\mu(aDL_1(RAN, GTH))$
13.5	17.0	16.7	19.8

Table 42: Twenty three-sign phrases for CopyCat suggested by this dissertation

Index	Phrase	Cost
1	CAT BEHIND BOX	58.84
2	CAT BEHIND WALL	67.56
3	CAT ON BOX	69.42
4	CAT IN BOX	73.44
5	CAT ON WALL	74.38
6	ALLIGATOR BEHIND BOX	137.00
7	CAT UNDER WAGON	139.13
8	ALLIGATOR BEHIND WALL	145.73
9	SNAKE BEHIND BOX	148.30
10	CAT IN WAGON	149.71
11	SPIDER ON BOX	149.95
12	CAT IN FLOWERS	153.58
13	SNAKE ON BOX	153.98
14	ALLIGATOR ON BOX	154.38
15	SPIDER ON WALL	154.91
16	CAT BEHIND CHAIR	156.67
17	SNAKE BEHIND WALL	157.02
18	CAT ON BED	158.11
19	ALLIGATOR IN BOX	159.83
20	SPIDER IN BOX	162.55

verification accuracy in suggesting phrases. However, should the game require more variety, the game designer could either easily pick other phrases from the ranking by specifying a larger k , or modify the Yen’s algorithm to skip certain sign combinations after $k_0 < k$ paths have been generated. This flexibility allows a game designer to conveniently trade verification accuracy with phrase variety in a scalable way.

Table 43: Twenty four-sign phrases for CopyCat suggested by this dissertation

Index	Phrase	Cost
21	CAT UNDER BLUE BOX	75.69
22	CAT UNDER GREEN BOX	76.19
23	CAT BEHIND GREEN BOX	84.07
24	CAT UNDER BLUE WALL	84.36
25	CAT UNDER BLACK WALL	84.39
26	CAT IN GREEN BOX	86.38
27	CAT UNDER GREEN WALL	91.07
28	CAT IN BLUE BOX	91.46
29	CAT UNDER ORANGE WALL	94.66
30	CAT IN WHITE WALL	96.04
31	CAT ON BLUE BOX	97.89
32	CAT ON GREEN BOX	98.94
33	CAT BEHIND GREEN WALL	98.95
34	CAT IN BLUE WALL	100.13
35	CAT BEHIND ORANGE WALL	102.80
36	CAT ON WHITE WALL	105.23
37	CAT ON ORANGE WALL	107.60
38	CAT BEHIND BLUE BOX	110.28
39	CAT BEHIND BLACK WALL	110.86
40	CAT BEHIND BLUE WALL	118.95

Table 44: Nineteen five-sign phrases for CopyCat suggested by this dissertation

Index	Phrase	Cost
41	BLUE ALLIGATOR BEHIND GREEN BOX	97.32
42	BLUE ALLIGATOR UNDER GREEN BOX	105.91
43	BLUE ALLIGATOR IN GREEN BOX	107.86
44	BLUE ALLIGATOR BEHIND GREEN WALL	112.21
45	BLUE ALLIGATOR UNDER BLACK WALL	114.11
46	BLUE ALLIGATOR BEHIND ORANGE WALL	116.05
47	BLUE SPIDER IN GREEN BOX	116.28
48	BLUE SPIDER UNDER GREEN BOX	116.75
49	BLUE ALLIGATOR IN WHITE WALL	117.52
50	BLUE ALLIGATOR ON GREEN BOX	118.99
51	BLUE SPIDER ON GREEN BOX	120.26
52	BLUE ALLIGATOR UNDER GREEN WALL	120.79
53	BLUE ALLIGATOR BEHIND BLACK WALL	124.11
54	BLUE ALLIGATOR UNDER ORANGE WALL	124.38
55	BLUE SPIDER UNDER BLACK WALL	124.95
56	BLUE ALLIGATOR ON WHITE WALL	125.28
57	BLUE SPIDER IN WHITE WALL	125.94
58	BLUE SPIDER ON WHITE WALL	126.55
59	BLUE ALLIGATOR ON ORANGE WALL	127.65

Table 45: Twenty three-sign phrases for CopyCat-Adult suggested by this dissertation

Index	Phrase	Cost
1	CAT ON BOX	68.10
2	CAT ON WALL	68.60
3	CAT BEHIND WALL	70.91
4	CAT BEHIND BOX	71.21
5	CAT IN BOX	89.09
6	CAT BEHIND BED	129.68
7	CAT BEHIND FLOWERS	130.18
8	SPIDER ON BOX	137.00
9	SPIDER ON WALL	137.51
10	SNAKE ON BOX	139.12
11	CAT BEHIND CHAIR	139.53
12	SNAKE ON WALL	139.62
13	CAT IN FLOWERS	141.25
14	SNAKE BEHIND WALL	145.18
15	SNAKE BEHIND BOX	145.49
16	ALLIGATOR BEHIND WALL	147.34
17	ALLIGATOR BEHIND BOX	147.65
18	CAT ON BED	148.25
19	CAT IN WAGON	148.93
20	ALLIGATOR ON BOX	151.08

Table 46: Twenty four-sign phrases for CopyCat-Adult suggested by this dissertation

Index	Phrase	Cost
21	CAT IN BLUE BOX	87.39
22	CAT UNDER BLUE BOX	88.13
23	CAT IN WHITE WALL	89.02
24	CAT ON BLUE BOX	91.19
25	CAT UNDER GREEN WALL	93.01
26	CAT BEHIND GREEN WALL	93.49
27	CAT IN GREEN WALL	93.63
28	CAT UNDER GREEN BOX	93.86
29	CAT BEHIND GREEN BOX	94.33
30	CAT BEHIND BLACK WALL	94.78
31	CAT IN ORANGE WALL	94.86
32	CAT IN BLUE WALL	97.71
33	CAT UNDER BLUE WALL	98.45
34	CAT ON GREEN WALL	101.97
35	CAT ON GREEN BOX	102.82
36	CAT UNDER ORANGE WALL	103.88
37	CAT BEHIND BLUE BOX	103.93
38	CAT ON WHITE WALL	105.85
39	CAT ON ORANGE WALL	106.37
40	CAT UNDER BLACK WALL	108.04

Table 47: Nineteen five-sign phrases for CopyCat-Adult suggested by this dissertation

Index	Phrase	Cost
41	BLUE ALLIGATOR UNDER GREEN WALL	107.29
42	BLUE ALLIGATOR UNDER GREEN BOX	108.14
43	BLUE ALLIGATOR IN WHITE WALL	110.52
44	BLUE ALLIGATOR BEHIND GREEN WALL	112.10
45	BLUE ALLIGATOR BEHIND GREEN BOX	112.95
46	BLUE ALLIGATOR BEHIND BLACK WALL	113.39
47	BLUE ALLIGATOR IN GREEN WALL	115.13
48	BLUE ALLIGATOR IN GREEN BOX	115.98
49	BLUE ALLIGATOR IN ORANGE WALL	116.36
50	BLUE ALLIGATOR UNDER ORANGE WALL	118.16
51	BLUE ALLIGATOR UNDER BLACK WALL	122.32
52	BLUE SPIDER IN WHITE WALL	124.48
53	BLUE SPIDER UNDER GREEN WALL	124.66
54	BLUE SPIDER UNDER GREEN BOX	125.51
55	BLUE ALLIGATOR BEHIND ORANGE WALL	126.89
56	BLUE ALLIGATOR ON GREEN WALL	127.14
57	BLUE ALLIGATOR ON GREEN BOX	127.99
58	BLUE SPIDER IN GREEN WALL	129.09
59	BLUE SPIDER IN GREEN BOX	129.94

CHAPTER VII

DISCUSSION AND FUTURE WORK

Discriminative algorithms are designed to optimize for *differences* in data. They are sensitive to noise, especially label noise [109]. Furthermore, segmental labels (states), unavailable in sequence classification, are inferred indirectly by some initial modeling of the training data. For example, segments of data are assigned to each state of an HMM in an unsupervised manner (EM) during training. The boundary of these segments can vary significantly if the training data are noisy or indistinct. The variance of signing also reduces the reliability of the initial estimation, which affects the accuracy of segmental discriminative analysis. For CopyCat data used in this dissertation, we have identified a few major sources of variance/noise as follows:

- Multiple users, especially child users
- Hand-dominance, including mixed and switching hand-dominance
- Different familiarity with ASL and the game
- Inaccurate labeling from inaccurate segmentation of the sequence (partly caused by the problems listed above)

Our SBHMMs showed significant improvement in accuracy over traditional HMMs and other current technologies in several domains. However, SBHMMs had significantly lower performance with the full set of CopyCat data. We believe this is mainly due to the variance/noise in the data, because testing on one child and the cleaner CopyCat-Adult data did show the trend of improvement. Further inspection also shows that the data cloud of many classes are inseparable even to a human, which suggests future improvement by including better input features and using a stronger dynamic model. More accurate labeling and segmentation should further reduce data variance of the same user. In the future,

we would like to promote user invariance through creating user independent features and preprocessing data (*e.g.*, maximum likelihood linear transformation, MLLT), so that the discriminative analysis results can be more reliable for feature selection. We strongly believe that including handshape information will greatly improve the results. As a first step, the CopyCat team has computed several handshape features as shown in Figure 19 [113]. In addition, we can introduce more tokens in training and recognition to address variance between users as well as some disfluencies. Discriminative clustering methods, such as DISC proposed in this dissertation, can reduce the requirement on training data as the number of tokens increases. However, generally speaking, we still need to collect more data in order to improve accuracy through these efforts.

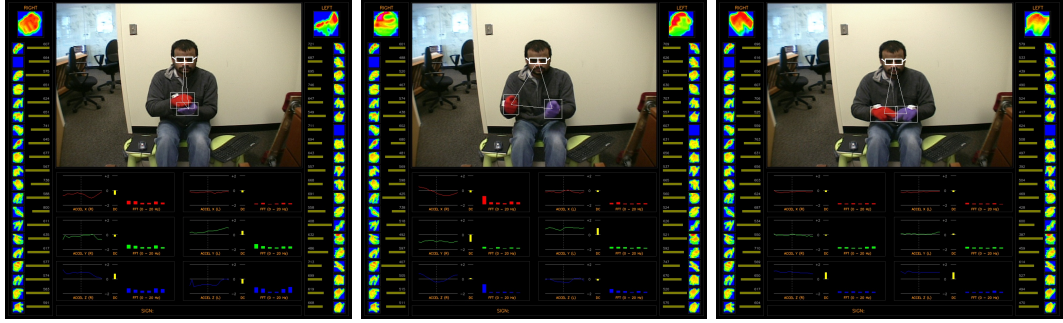


Figure 19: The new handshape features (courtesy to Zahoor Zafrulla) for ALLIGATOR(left) BEHIND(middle) WALL(right).

Generally speaking, signing features such as the length of arms of adult participants differ from those of children. Because the CopyCat-Adult data are higher quality, we are also investigating if we can introduce normalization to merge the CopyCat-Adult data with the CopyCat data. Another way to address data variance is to collect more data. As the amount of data grows, the impact of random noise starts to fade.

We have started deploying a trained automatic verifier to the real environment to replace the human verifier. As the project CopyCat proceeds to the next stage, we would like to evaluate the testing accuracy of some unseen phrases suggested by the proposed technique using Yen’s algorithm and the BIG/UNG predictor(s). The idea of modifying of the Yen’s algorithm to improve variety in Section 6.7.3 can be also validated.

The CopyCat game contains a small vocabulary and very limited phrases, but we are confident in extending the proposed discriminative segmental analysis techniques to more complicated ASL recognition/verification applications with a larger vocabulary. First, different ASL verification tasks may have different “costs” for false positives and false negatives. Instead of manually searching the “best” trade-off, the probe technique is able to automatically provide a reasonable “initial guess” for parameter tuning. Second, segmental phrase analysis tools, such as BIG, are scalable with increasing vocabulary size. Third, DISC-SBHMMs help model complexity stay tractable by tying redundant states. For example, explicitly modeling disfluencies such as false starts, sneezes, coughs, and so on usually requires a very fine-scale labeling of data, which significantly increases the number of labels for the recognition system [13]. We believe that DISC-SBHMMs should be able to tie similar sub-sequences and remove unnecessary states from the expanded label set. In addition, when the vocabulary size becomes very large, generative algorithms that are relatively lighter in computation, such as the clustering trees used for Chinese Sign Language recognition [28], can preprocess the data for discriminative algorithms. Lastly, as the vocabulary grows, disambiguating minimal pairs becomes more and more important to attain high recognition accuracy. To disambiguate them, discriminative segmental feature selection can be more effective than generative models. However, for the segmental discriminative analysis techniques proposed in this dissertation, state labels, or sometimes even class labels, are estimated from a continuous input stream, and thus they may be inaccurate. Therefore, when increasing the size of vocabulary, we need to ensure that certain types of invariance (*e.g.*, within class and/or between users) exist in the dataset; otherwise, the algorithms can become confused due to random noise.

In the future, we would also like to apply the proposed segmental discriminative analysis to other sequence classification tasks, especially those with temporal structures. We believe that the improvement in performance will be similar to that of the ASLR algorithms and other example domains discussed in this work.

CHAPTER VIII

CONCLUSION

This dissertation has presented segmental discriminative analysis techniques for American Sign Language recognition and other sequence classification tasks. By grouping individual frames into segments, SBHMM is able to automatically select discriminative features that are only informative for part of the sequence, and DISC is able to discover the common sub-signs to reduce modeling complexity. In contrast, by decomposing an entire temporal sequence into segments, BIG is able to achieve scalability with strong prediction capabilities that are comparable to the best training results that require intractable computation. Experimental results, mostly in ASL recognition/verification but also in various other applications, show that the proposed segmental discriminative analysis techniques are able to achieve high accuracy at manageable computational cost.

In summary, the proposed segmental discriminative analysis techniques include the following capabilities:

1. A discriminative feature selection algorithm for HMMs (SBHMMs).
2. A discriminative pattern discovery algorithm for HMMs (DISC).
3. A parsimonious representation of a given vocabulary of signs for machine perception.
4. A phrase selection method for sign language verification tasks using discriminative analysis (BIG).

APPENDIX A

BASICS OF LINGUISTICS

A.1 Linguistics of Spoken Language

In modern linguistics, “grammar” is the set of rules and elements, including *phonology*, *morphology*, *syntax*, and so on, that make up a language [97]. Phonology studies the way words are pronounced. For instance, the English word “bat” is pronounced with *phonemes* /b/, /æ/, and /t/. A phoneme is the smallest contrastive unit in a *minimal pair*: two different words that are otherwise identical except for one aspect of their production. Furthermore, these aspects, or linguistic *features* [51] characterize the utterance, and thus can measure the phonetic difference/similarity. Table 4 in Chapter 4 shows the contrast of /p/ and /b/. Morphology studies the way words are formed from “smaller meaning bearing units.” For instance, “unable” is formed by two *morphemes*: an *affix* “un-” and a *stem* “able.” Syntax studies the way sentences are composed by words. For instance, ‘I have had lunch’ includes the subject “I,” the verb “have,” and the object “lunch.”

A.2 Linguistics of Sign Language

ASL is compositional and it can be broken down into smaller “basic units” as any spoken language, first shown by Stokoe’s pioneering work [94] in 1960. Since then, different “basic units” of ASL phonemes have been proposed. The *Stokoe notation* describes ASL according to its three major formational categories: handshapes (*dez*), locations (*tab*), and movements (*sig*). However, these descriptors, which represent a sign as one simultaneous bundle of linguistic features, inadequately capture sequential internal segments [95]. The first detailed model of ASL capable of representing sequential segments was the movement-hold model, proposed by Liddell and Johnson [56]. This model describes ASL according to two types of sequentially ordered segments: movement segments (M) and hold segments (H). The third model commonly used for ASL is the hand-tier model [82] based “on the fact that most

signs are characterized by a single hand configuration in which no (linguistic) features vary throughout the sign.”¹ This model claims that the three major formational categories are handshapes, locations, and movements as the Stokoe’s model, which are similar to vowels, consonants, and tone in spoken languages, respectively [25]. However, while similar to the movement-hold model, the hand-tier model organizes locations and movements sequentially, the hand configuration typically characterizes the whole sequence simultaneously: that is, only the changing linguistic features are represented sequentially for the purpose of preventing duplication. Beside the phonological motivations, the hand-tier model representation also introduces morphological benefits.

¹Some “hand shape changes” in signs are actually finger position changes, claimed by Sandler [82].

APPENDIX B

STATISTIC TESTS USED IN THIS DISSERTATION

B.1 Significance Test

In hypothesis testing [101], a *null hypothesis*, H_0 is to be tested. For example, “the GRA compensation C_{GT} for Georgia Tech student equals the compensation C_O for students in other public school,” and an *alternative hypothesis*, H_a is literally the alternative to the null hypothesis. In the previous example, the alternative hypothesis can be “ C_{GT} differs from C_O ,” which is a “two-sided test.” Other alternative hypotheses include “ $C_{GT} < C_O$ ” and “ $C_{GT} > C_O$,” which are one-sided tests. The range of the *test statistic*, such as $\frac{C_{GT}-C_O}{\sigma}$, is divided into the *acceptance region* and the *rejection region* for the null hypothesis separated by *critical values*. For example, if the critical value is -2, when $\frac{C_{GT}-C_O}{\sigma} < -2$, we reject the null hypothesis. During the setup of the critical value, two types of errors can occur. A *Type I* error rejects the null hypothesis by mistake, and a *Type II* accepts the null hypothesis by error. We can trade Type I errors with Type II errors by setting the critical value at a different level. This setting of the critical value determines the *significance level* α and vice versa. Significance level α is the probability that we will make type I errors. If $\alpha = 0.05$ and we reject the null hypothesis, we are $(1 - 0.05) * 100\% = 95\%$ confident that the alternative hypothesis is true.

B.2 Paired t-test

Paired t-test [101] measures the statistical significance of the difference between the population mean of two set of measured value, $\{x_1, x_2, \dots, x_n\}$ and $\{y_1, y_2, \dots, y_n\}$ on the same set of samples. The paired t-test uses correlation between pairs to reduce the variance and thus has additional statistical power. The test statistic of a paired t-test is mathematically defined as:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}},$$

in which \bar{d} is the mean of the paired difference ($x_i - y_i$), s_d is the standard deviation of the paired difference, and n is the sample size. The p-value for null hypothesis (the probability that the population means of x_i and y_i are equal) can be found in a t-distribution table.

B.3 Tests for Gaussian Distribution

Most of the quantitative tests listed here require the computation of the central moments.

The k -th order central moments M_k are defined by $M_k = \frac{\sum (X - \bar{X})^k}{N}$

$$Skew = \frac{M_3}{M_2^{3/2}}$$

Skewness measures the symmetry of a distribution. The Gaussian distribution and other symmetric distributions have a skewness of zero.

$$Kurtosis = \left(\frac{M_4}{M_2^2} \right) - 3$$

Kurtosis measures the “peakiness” of the distribution. The original kurtosis has a systematic bias of 3. The formula above is the unbiased kurtosis, which is used in this dissertation. The Gaussian distribution has a kurtosis of zero. If the kurtosis is less than zero, the distribution is “too flat,” defined as *platykurtic*. For example, $Kurtosis(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) = -1.22$. A kurtosis is greater than zero is defined as *leptokurtic*. The leptokurtic distribution can be explained as either a higher frequency than the Gaussian distribution around the mean or a thicker tail than the Gaussian. For example, $Kurtosis(0, 0, 0, 0, 0, 0, 0, 0, 0, 1) = 5.11$.

$$Z = \frac{Skew}{\sqrt{6/N}}$$

The z-test can measure whether a distribution differs significantly from the Gaussian distribution. For the above formula, the Gaussian distribution will have a z-test value of 0. The farther the z-test value is from zero, the further the data distribution departs from the true Gaussian distribution.

$$KS = \max |F_0(X) - S_N(X)|$$

The Kolmogorov-Smirnov (KS) test measures the maximum difference between the accumulated probability of the theoretical distribution $F_0(X)$ and the real distribution $S_N(X)$. Here, $F_0(X) = 1 - \text{erf}(X)$, and the data distribution will have a KS value closer to zero if it is more “Gaussian.”

REFERENCES

- [1] *Allied Tactical Publication, Allied Maritime Maneuvering Instructions, ATP1*, vol. 2. NATO, 1983.
- [2] ALLWEIN, E., SCHAPIRE, R., and SINGER, Y., “Reducing multiclass to binary: A unifying approach for margin classifiers,” *Journal of Machine Learning Research*, vol. 1, pp. 113–141, 2001.
- [3] ALON, J., *Spatiotemporal Gesture Segmentation*. PhD thesis, Boston University, 2006.
- [4] ANDERSON, O., DALSGAARD, P., and BARRY, W., “On the use of data-driven clustering technique for identification of poly- and mono-phonemes for four European languages,” in *Proc. of IEEE ICASSP*, pp. 121–124, 1994.
- [5] ANDREWS, J., VERNON, M., and LAVIGNE, M., “The deaf suspect/defendant and the bill of rights,” *Views*, 2006.
- [6] BAHL, L., BROWN, P., SOUZA, P., and MERCER, R., “Acoustic Markov models used in the Tangora speech recognition system,” in *Proc. of IEEE ICASSP*, 1988.
- [7] BAUER, B. and KRAISS, K., “Towards an automatic sign language recognition system using subunits,” *LNAI*, vol. 2298, pp. 64–75, 2002.
- [8] BESAG, J., “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society*, pp. 259–302, 1986.
- [9] BISHOP, C., *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [10] BLACK, M., TEPPERMAN, J., LEE, S., PRICE, P., and NARAYANAN, S., “Automatic detection and classification of disfluent reading miscues in young children’s speech for the purpose of assessment,” in *Proc. of InterSpeech-ICSLP*, pp. 206–209, 2007.
- [11] BRASHEAR, H., PARK, K.-H., LEE, S., HENDERSON, V., HAMILTON, H., and STARNER, T., “American Sign Language recognition in game development for deaf children,” in *Proc. of ACM SIGACCESS Conference on Computers and Accessibility*, pp. 75–86, 2006.
- [12] BRASHEAR, H., STARNER, T., LUKOWICZ, P., and JUNKER, H., “Using multiple sensors for mobile sign language recognition,” in *Proc. of IEEE ISWC*, 2003.
- [13] BRASHEAR, H., *Improving the Efficacy of Automated American Sign Language Practice Tools*. PhD thesis, Proposal, Georgia Institute of Technology, 2007.
- [14] BREGLER, C., COVELL, M., and SLANEY, M., “Video rewrite: Driving visual speech with audio,” in *Proc. of ACM SIGGRAPH*, pp. 353–360, 1997.
- [15] CHEN, Y., “Chinese Sign Language recognition and synthesis,” in *Proc. of IEEE International workshop on AMFG*, 2003.

- [16] CHIU, B., KEOGH, E., and LONARDI, S., “Probabilistic discovery of time series motifs,” in *Proc. of the ninth ACM SIGKDD*, pp. 493–498, 2003.
- [17] COOPER, M. and FOOTE, J., “Summarizing popular music via structural similarity analysis,” in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.
- [18] DENG, L., DROPO, J., and ACERO, A., “Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition,” *IEEE Trans. on Speech and Audio Processing*, vol. 11, no. 6, pp. 569–580, 2003.
- [19] DENG, L., WU, J., DROPO, J., and ACERO, A., “Analysis and comparison of two speech feature extraction/compensation algorithms,” *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 477–479, 2005.
- [20] DIETTERICH, T. G., “Machine learning for sequential data: A review,” *LNCS*, vol. 2396, pp. 15–30, 2002.
- [21] DIJKSTRA, E. W., “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [22] DROPO, J. and ACERO, A., “Experimenting with a global decision tree for state clustering in automatic speech recognition systems,” in *IEEE ICASSP*, 2009.
- [23] DUDA, R., HART, P., and STORK, D., *Pattern Classification*. Wiley-Interscience, second ed., 2000.
- [24] EHSANI, F. and KNODT, E., “Speech technology in computer-aided language learning: Strengths and limitations of a new call paradigm,” *Language Learning and Technology*, vol. 2, no. 1, pp. 45–60, 1998.
- [25] EMMOREY, K., *Language, Cognition, and the Brain – insights from sign language research*. Lawrence Erlbaum Associates, 2002.
- [26] EPPSTEIN, D., “Finding the k shortest paths,” *SIAM Journal on Computing*, vol. 28, pp. 652–673, 1998.
- [27] ESKENAZI, M., “Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype,” *Language Learning and Technology*, vol. 2, no. 2, pp. 62–76, 1999.
- [28] FANG, G., GAO, W., and ZHAO, D., “Large sign vocabulary sign recognition based on hierarchical decision tree,” in *Proc. of ICMI*, 2003.
- [29] FREUND, Y. and SCHAPIRE, R., “A decision theoretic generalization of on-line learning and application to boosting,” *Journal of Computer and System Science*, vol. 55(1), pp. 119–139, 1995.
- [30] GUNAWARDANA, A., MAHAJAN, M., ACERO, A., and PLATT, J., “Hidden conditional random fields for phone classification,” in *Proc. of International Conference on Speech Communication and Technology*, 2005.

- [31] HAMERS, J., “Cognitive and language development of bilingual children,” in *Cultural and language diversity and the deaf experience* (PARASINIS, L., ed.), pp. 51–75, Cambridge University Press, 1998.
- [32] HAMID, R., MADDI, S., BOBICK, A., and ESSA, I., “Structure from statistics - unsupervised activity analysis using suffix trees,” in *Proc. of IEEE ICCV*, 2007.
- [33] HAN, J., AWAD, G., and SUTHERLAND, A., “Modelling and segmenting subunits for sign language recognition based on hand motion analysis,” *Pattern Recognition Letters*, vol. 30, no. 6, pp. 623–633, 2009.
- [34] HERMANSEY, H., ELLIS, D. P., and SHARMA, S., “Tandem connectionist feature extraction for conventional HMM systems,” in *Proc. of IEEE ICASSP*, pp. 1635–1638, 2000.
- [35] HOLT, G. T., HENDRIKS, P., and ANDRINGA, T., “Why don’t you see what I mean? prospects and limitations of current automatic sign recognition research,” *Sign Language Studies*, vol. 6, pp. 416–437, 2006.
- [36] JAAKKOLA, T. S. and HAUSSLER, D., “Exploiting generative models in discriminative classifiers,” in *Proc. of NIPS*, 1999.
- [37] JÄRVELIN, K. and KEKÄLÄINEN, J., “Cumulated gain-based evaluation of IR techniques,” *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.
- [38] JEHAN, T., “Perceptual segment clustering for music description and time-axis redundancy cancellation,” in *Proc. of ICMIR*, 2004.
- [39] JING, Y., PAVLOVIC, V., and REHG, J., “Efficient discriminative learning of Bayesian network classifier via boosted augmented naive bayes,” in *Proc. of ICML*, pp. 369–376, 2005.
- [40] JOHNSON, R., LIDDELL, S., and ERTING, C., “Unlocking the curriculum: principles for achieving access in deaf education.” Gallaudet Research Institute Working Paper, 1989.
- [41] JUANG, B. and KATAGIRI, S., “Discriminative learning for minimum error classification,” *IEEE Trans. on Signal Processing*, vol. 40, pp. 119–139, 1992.
- [42] JUANG, B. and RABINER, L., “The segmental k-means algorithm for estimating parameters of hidden Markov models,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1639–1641, 1990.
- [43] KANNAN, A., WINN, J., and ROTHER, C., “Clustering appearance and shape by learning jigsaws,” in *Proc. of NIPS*, vol. 19, 2006.
- [44] KANNAPELL, B., “An examination of deaf college students’ attitude toward ASL and English,” in *The sociolinguistics of the deaf community* (LUCAS, C., ed.), pp. 191–210, Academic Press, 1989.
- [45] KENDALL, M. and GIBBONS, J. D., *Rank Correlation Methods*. A Charles Griffin Title, 5 ed., September 1990.

- [46] KEOGH, E., LIN, J., and TRUPPEL, W., "Clustering of time series subsequences is meaningless: implications for previous and future research," in *Proc. of IEEE ICDM*, pp. 115–122, 2003.
- [47] KIM, M. and PAVLOVIC, V., "Discriminative learning of mixture of Bayesian network classifiers for sequence classification," in *Proc. of IEEE CVPR*, pp. 268–275, 2006.
- [48] KRAMER, J. and LEIFER, L., "The talking glove: An expressive and receptive 'verbal' communication aid for the deaf, deaf-blind, and nonvocal," in *Proc. of Conference on Computer Technology, Special Education, and Rehabilitation*, 1987.
- [49] LAFFERTY, J., MCCALLUM, A., and PEREIRA, F., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of 18th ICML*, pp. 282–289, 2001.
- [50] LANE, H., BOYES-BRAEM, P., and BELLUGI, U., "Preliminaries to a distinctive feature analysis of handshapes in American Sign Language," *Cognitive Psychology*, vol. 8, pp. 263–289, 1976.
- [51] LAVER, J., *Principles of phonetics*. Cambridge University Press, 1994.
- [52] LEE, C. H., "A tutorial on speaker and speech verification," in *Proc. of NORSIG*, pp. 9–16, 1998.
- [53] LEE, K., HON, H., and REDDY, R., "An overview of the SPHINX speech recognition system," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38(1), pp. 35–45, 1990.
- [54] LEE, S., HENDERSON, V., BRASHEAR, H., STARNER, T., HAMILTON, H., and HAMILTON, S., "User-centered development of a gesture-based American Sign Language (ASL) game," in *Proc. of Instructional Technology and Education of the Deaf Symposium*, 2005.
- [55] LIDDELL, S. K., "Think and believe: Sequentiality in American Sign Language," *Language*, vol. 60, no. 2, pp. 372–399, 1984.
- [56] LIDDELL, S. K. and JOHNSON, R. E., "American Sign Language: The phonological base," *Sign Language Studies*, vol. 64, pp. 195–277, 1989.
- [57] LOEDING, B., SARKAR, S., PARASHAR, A., and KARSHMER, A., "Progress in automated computer recognition of sign language," *LNCIS*, vol. 3118, pp. 1079–1087, 2004.
- [58] LV, F. and NEVATIA, R., "Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost," in *Proc. of IEEE ECCV*, pp. IV: 359–372, 2006.
- [59] MAKHOUL, J. and ROUCOS, S., "Vector quantization in speech coding," *Proc. of IEEE*, vol. 73, no. 21, 1985.
- [60] MARTINS, E. and PASCOAL, M., "A new implementation of Yen's ranking loopless paths algorithm," 2000.

- [61] MCGUIRE, R., HERNANDEZ-REBOLLAR, J., STARNER, T., HENDERSON, V., BRASHEAR, H., and ROSS, D., “Towards a one-way american sign language translator,” in *IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pp. 620–625, 2004.
- [62] MEYER, C., “Towards ‘large margin’ speech recognizers by boosting and discriminative learning,” in *Proc. of ICML*, pp. 419–426, 2002.
- [63] MEYER, C., “Utterance-level boosting of HMM speech recognizers,” in *Proc. of IEEE ICASSP*, 2002.
- [64] MINNEN, D., ESSA, I., ISBELL, C., and STARNER, T., “Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery,” in *Proc. of IEEE ICDM*, 2007.
- [65] MITCHELL, R. E. and KARCHMER, M. A., “Chasing the mythical ten percent: parental hearing status of deaf and hard of hearing students in the United States,” *Sign Language Studies*, vol. 4, no. 2, pp. 138–163, 2004.
- [66] MORENCY, L.-P., QUATTONI, A., and DARRELL, T., “Latent-dynamic discriminative models for continuous gesture recognition,” in *IEEE CVPR*, pp. 1–8, 2007.
- [67] MOSTOW, J., “Is ASR accurate enough for automated reading tutors, and how can we tell?,” in *Proc. of InterSpeech-ICSLP*, pp. 837–840, 2006.
- [68] MOSTOW, J., “Experience from a reading tutor that listens: Evaluation purposes, excuses, and methods,” in *Interactive Literacy Education: Facilitating Literacy Environments Through Technology* (KINZER, C. K. and VERHOEVEN, L., eds.), pp. 117 – 148, New York: Erlbaum Publishers, 2008.
- [69] MURPHY, K., “Hidden semi-Markov models,” 2002.
- [70] NAYAK, S., *Representation of Learning for Sign Language Recognition*. PhD thesis, University of South Florida, 2008.
- [71] NAYAK, S., SARKAR, S., and LOEDING, B., “Automated extraction of signs from continuous sign language sentences using iterated conditional modes,” in *IEEE CVPR*, pp. 2583–2590, 2009.
- [72] OH, S. M., *Switching linear dynamic systems with higher-order temporal structure*. PhD thesis, Georgia Institute of Technology, 2009.
- [73] ONG, C. and RANGANATH, S., “Automatic sign language analysis: A survey and the future beyond lexical meaning,” *IEEE Trans. on PAMI*, vol. 27, no. 6, pp. 873–891, 2005.
- [74] PAVLOVIC, V., “Model-based motion clustering using boosted mixture modeling,” in *Proc. of IEEE CVPR*, pp. I:811–818, 2004.
- [75] PAVLOVIC, V., REHG, J., and MACCORMICK, J., “Learning switching linear models of human motion,” in *Proc. of NIPS*, 2001.

- [76] POTAMIANOS, G., NETI, C., GRAVIER, G., and GARG, A., “Automatic recognition of audio-visual speech: Recent progress and challenges,” *Proc. of the IEEE*, vol. 91, no. 9, 2003.
- [77] POVEY, D., KINGSBURY, B., MANGU, L., SAON, G., SOLTAU, H., and ZWEIG, G., “fMPE: Discriminatively trained features for continuous speech recognition,” in *Proc. of IEEE ICASSP*, 2005.
- [78] QUATTONI, A., WANG, S., MORENCY, L., COLLINS, M., and DARRELL, T., “Hidden conditional random fields,” *IEEE Trans. on PAMI*, vol. 29, no. 10, pp. 1848–1852, 2007.
- [79] RABINER, L., “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. of The IEEE*, vol. 77(2), pp. 257–286, 1989.
- [80] RABINER, L. and JUANG, B., *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ, Printice Hall, 1993.
- [81] RUSSEL, M. and COOK, A., “Experimental evaluation of duration modelling techniques for automatic speech recognition,” in *Proc. of IEEE ICASSP*, pp. 2376–2379, 1987.
- [82] SANDLER, W., “The spreading hand autosegment of American Sign Language,” *Sign Language Studies*, vol. 50, pp. 1–28, 1986.
- [83] SANDLER, W., *Phonological Representation of the Sign*. Dordrecht: Foris, 1989.
- [84] SANDLER, W. and LILLO-MARTIN, D., *Sign Language and Linguistic Universals*. Cambridge University Press, 2006.
- [85] SARAWAGI, S. and COHEN, W., “Semi-Markov conditional random fields for information extraction,” in *Proc. of NIPS*, 2004.
- [86] SCHAPIRE, R., FREUND, Y., BARTLETT, P., and LEE, W., “Boosting the margin: a new explanation for the effectiveness of voting methods,” *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.
- [87] SHA, F., “Shallow parsing with conditional random fields,” in *Proc. of HLT-NAACL*, 2003.
- [88] SHA, F. and SAUL, L., “Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models,” in *Proc. of IEEE ICASSP*, pp. IV 313–316, 2007.
- [89] SHA, F. and SAUL, L., “Large margin hidden Markov models for automatic speech recognition,” in *Proc. of NIPS*, 2007.
- [90] SMITH, P. and N. LOBO, M. S., “Temporalboost for event recognition,” in *Proc. of ICCV*, pp. 733–740, 2005.
- [91] SPENCER, P., “Communication behaviors of infants with hearing loss and their hearing mothers,” *Journal of Speech and Hearing Research*, vol. 36, pp. 311–321, 1993.

- [92] SPENCER, P., “The expressive communication of hearing mothers and deaf infants,” *American Annals of the Deaf*, vol. 138, pp. 275–283, 1993.
- [93] STARNER, T., WEAVER, J., and PENTLAND, A., “Real-time American Sign Language recognition using desk and wearable computer based video,” *IEEE Trans. on PAMI*, vol. 20, no. 12, pp. 1371–1375, 1998.
- [94] STOKOE, W. C., “Sign language structure: An outline of the visual communication systems of the American deaf,” *Studies in Linguistics*, vol. 8, 1960.
- [95] SUPALLA, T. and NEWPORT, E., “How many seats in a chair? The derivation of nouns and verbs in American Sign Language,” in *Understanding Language Through Sign Language Research* (SIPLE, ed.), pp. 91–132, New York:Academic Press, 1978.
- [96] TANAWONGSUWAN, R. and BOBICK, A., “Performance analysis of time-distance gait parameters under different speeds,” *LNCS*, vol. 2688, pp. 1060–1068, 2003.
- [97] VALLI, C. and LUCAS, C., *Linguistics of American Sign Language: An introduction*. Gallaudet University Press, third ed., 2000.
- [98] VOGLER, C. and METAXAS, D., “A framework for recognizing the simultaneous aspects of American Sign Language,” *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 358–384, 2001.
- [99] WALDRON, M. and KIM, S., “Isolated ASL sign recognition system for deaf persons,” *IEEE Transactions on Rehabilitation Engineering*, vol. 3, pp. 261–271, 1995.
- [100] WANG, C., GAO, W., and SHAN, S., “An approach based on phonemes to large vocabulary Chinese Sign Language recognition,” in *Proc. of Automatic Face and Gesture Recognition*, pp. 411 – 416, 2002.
- [101] WEISS, N., *Introductory Statistics*. Addison Wesley, 2007.
- [102] WESTEYN, T., BRASHEAR, H., ATRASH, A., and STARNER, T., “Georgia Tech Gesture Toolkit: Supporting experiments in gesture recognition,” in *Proc. of ICMI*, pp. 85–92, 2003.
- [103] WOODLAND, P., ODELL, J., VALTCHEV, V., and YOUNG, S., “Large vocabulary continuous speech recognition using HTK,” in *Proc. of IEEE ICASSP*, pp. II:125–128, 1994.
- [104] WOODLAND, P. and POVEY, D., “Large scale discriminative training for speech recognition,” in *Proc. of the Workshop on Automatic Speech Recognition*, 2000.
- [105] XU, L., VARADHARAJAN, V., MARAVICH, J., TONGIA, R., and MOSTOW, J., “De-SIGN: An intelligent tutor to teach American Sign Language,” in *Proc. of the SLATE Workshop on Speech and Language Technology in Education*, 2007.
- [106] YANG, H., SCLAROFF, S., and LEE, S., “Sign language spotting with a threshold model based on conditional random fields,” *IEEE Trans. on PAMI*, vol. 31, pp. 1264–1277, July 2009.
- [107] YEN, J., “Finding the k shortest loopless paths in a network,” *Management Science*, vol. 18, pp. 712–716, 1971.

- [108] YIN, P., CRIMINISI, A., WINN, J., and ESSA, I., “Tree-based classifiers for bilayer video segmentation,” in *Proc. of IEEE CVPR*, 2007.
- [109] YIN, P., CRIMINISI, A., WINN, J., and ESSA, I., “Bilayer segmentation of webcam videos using tree-based classifiers,” *IEEE Trans. on PAMI*, 2010.
- [110] YIN, P., ESSA, I., and REHG, J., “Asymmetrically boosted HMM for speech reading,” in *Proc. of IEEE CVPR*, pp. II:755–761, 2004.
- [111] YIN, P., ESSA, I., STARNER, T., and REHG, J., “Discriminative feature selection for hidden Markov models using segmental boosting,” in *Proc. of IEEE ICASSP*, 2008.
- [112] YIN, P., STARNER, T., HAMILTON, H., ESSA, I., and REHG, J., “Learning basic units in American Sign Language using discriminative segmental feature selection,” in *Proc. of IEEE ICASSP*, 2009.
- [113] ZAFRULLA, Z., STARNER, T., YIN, P., BRASHEAR, H., and HAMILTON, H., “American sign language phrase verification in an educational game,” in *submission to ICPR*, 2010.
- [114] ZUE, V. and GLASS, J., “Conversational interfaces: Advances and challenges,” *Proc. of the IEEE, Special Issue on Spoken Language Processing*, vol. 88, no. 8, pp. 1166–1180, 2000.